

Disseny d'aplicació multiplataforma per al sector hotel·ler



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Autor: Jordi Guiu Pujols

Director: Pau Fonseca (DEIO)

Data de defensa: 25/04/2019

Grau en enginyeria Informàtica

Enginyeria del Software

FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)

Agraïments

Vull donar àmpliament les gràcies a totes les persones que han participat directament o indirectament en aquest treball.

Al meu director Pau Fonseca i Casas per proposar-me aquest treball que m'ha permès descobrir noves tecnologies molt interessants i un nou món.

No podria presentar aquest treball sense donar les gràcies a la Mar Guinart que em va donar un cop de mà amb el disseny del icones de l'aplicació que van quedar tan bonics finalment. Va fer un treball excel·lent i mereix especial menció en aquests agraïments.

També donar moltes gràcies al Pau Romero per deixar-me el seu Mac tants cops com fes falta sense posar-m'hi cap problema perquè pogués provar l'aplicació que generava per a iOS. No podria oblidar-me tampoc de la Maria Macedo que em va deixar un iPhone desinteressadament perquè també pogués provar l'aplicació en un dispositiu Apple i pogués seguir endavant amb el treball. Gràcies per fer dels problemes simples tràmits.

Tampoc puc oblidar-me de donar les gràcies a tots els amics que m'han donat ànims constantment i sobretot a tots els companys i companyes dels Arreplegats de la Zona Universitària per ensenyar-me a tirar endavant amb tot allò que em proposo.

Finalment donar moltes gràcies a la meva mare i al meu pare per estar sempre darrere meu per donant-me una empenta i posar-me a lloc.

Resum

En un món cada vegada més globalitzat i tecnològic no tenim temps per pensar en com repercuteixen en el medi ambient les nostres accions. Un 40% de l'energia que es consumeix a Espanya, EUA o el Regne Unit prové dels edificis: segur que no estem malbaratant una enorme quantitat d'energia?

Existeixen diferents sistemes de gestió d'energia, però gaire bé no hi ha solucions que posin la informació a les mans de les persones. NECADA [1][2] que és un projecte de gestió d'energia desenvolupat en el InLab FIB busca l'optimització de l'energia tenint en compte factors ambientals, econòmics i socials.

Aquest treball pretén desenvolupar una aplicació multiplataforma per al sector hoteler, connectada a NECADA, que indiqui als clients dels hotels com poden estalviar energia de l'habitació on s'allotgen veient-se recompensats i empesos a contribuir al sistema.

Resumen

En un mundo cada vez más globalizado y tecnológico no tenemos tiempo para pensar en cómo repercuten en el medio ambiente nuestras acciones. Un 40% de la energía que se consume en España, EEUU o el Reino Unido proviene de los edificios: ¿Seguro que no estamos malgastando una enorme cantidad de energía?

Existen diferentes sistemas de gestión de energía, pero casi no hay soluciones que pongan la información en las manos de las personas. NECADA [1][2] que es un proyecto de gestión de energía desarrollado en el InLab FIB que busca la optimización de la energía teniendo en cuenta factores ambientales, económicos y sociales.

Este trabajo pretende desarrollar una aplicación multiplataforma para el sector hotelero, que se conecte a NECADA, e indique a los clientes de los hoteles como pueden ahorrar energía de la habitación donde se alojan viéndose recompensados y empujados a contribuir al sistema.

Abstract

In an increasingly globalized and technological world, we do not have time to think about how our actions affect our environment. 40% of the energy consumed in Spain, the USA or the United Kingdom comes from the buildings: surely we are not wasting a huge amount of energy?

There are different energy management systems, but there are no solutions that place information in the hands of people. NECADA [1][2] that is an energy management project developed at the InLab FIB that seeks optimization of energy taking into account environmental, economic and social factors.

This project intends to develop a multiplatform application for the hotel sector, connected to NECADA, which indicates to hotel customers how to save energy from the room where they are housed rewarding and encouraging them to contribute to the system.

Índex de continguts

Agraïments	1
Resum	2
Resumen	3
Abstract	4
1. Introducció i contextualització	15
1.1. Context	15
1.1.1. Introducció	15
1.1.2. Formulació del problema	15
1.2. Estat de l'art	16
1.2.1. L'estalvi energètic en la informàtica	16
1.2.2. Estudis de EMS	16
1.2.3. Les aplicacions gamificades	17
2. Abast	18
2.1. Objectius	18
2.2. Abast	19
2.2.1. Abast inicial	19
2.2.2. Abast final	19
2.3. Metodologia i rigor	20
2.3.1. Eines de desenvolupament	20
2.3.2. Eines de seguiment	20
2.3.3. Mètode de validació	21
3. Tecnologies, eines, conceptes i coneixements aplicats	22
3.1. Llenguatges de programació	22
3.1.1. C#	22
3.1.2. XAML	22
3.2. Visual Studio	22
3.3. Xamarin.Forms	23
3.3.1. Embedded Resource	24
3.4. MVVM	25
3.5. Llibreries de NuGet	25

3.5.1. FlowListView	26
3.5.2. PullToRefresh	26
3.5.3. Settings	26
3.5.1. JSON.NET	26
3.6. Android i iOS	26
4. Anàlisi de requisits i especificació	28
4.1. Stakeholders	28
4.1.1. Sector hotelier	28
4.1.2. Els clients dels hotels	28
4.1.3. Departament de Territori i Sostenibilitat	28
4.2. Requisits funcionals	29
4.3. Diagrama de casos d'ús	29
4.4. Casos d'ús	30
4.5. Requisits de qualitat	35
4.6. Esquema conceptual	38
4.6.1. Diagrama de classes UML	39
4.6.2. Explicació de les classes, atributs i associacions	39
5. Desenvolupament del projecte	41
5.1. Arquitectura del sistema	41
5.2. Disseny	42
5.2.1. Disseny de la interfície	43
5.2.1.1. Disseny de l'estructura	43
5.2.1.1.1. Vista principal	43
5.2.1.1.2. Vista de LogIn	45
5.2.1.1.3. Vista detall de factor	46
5.2.1.1.4. Vista detall de la situació	47
5.2.1.1.5. Vista de perfil	48
5.2.1.1.6. Vista de informació de l'aplicació	49
5.2.1.1.7. Vista splash	50
5.2.1.2. Colors	50
5.2.1.3. Icones	52
5.2.1.4. Tipografia	53
5.2.2. Disseny del software	54

5.2.3. Disseny de la BBDD	57
6. Planificació temporal	60
6.1. Descripció de les tasques	60
6.1.1. Treball inicial	60
6.1.2. Instal·lació i familiarització dels entorns	60
6.1.3. Familiarització amb Xamarin	60
6.1.4. Desenvolupament de l'aplicació .NET	61
6.1.5. Documentar el treball	61
6.1.6. Revisió i entrega de la versió final	61
6.2. Planificació inicial	61
6.2.1 Càrrega de treball de les tasques	61
6.2.2 Diagrama de GANTT	62
6.2.3. Valoració d'alternatives i pla d'acció	62
6.2.3.1. Planificació	62
6.2.3.2. Bugs	63
6.2.3.3. Errors de configuració	63
6.2.3.4. Errors de simulació	63
6.3. Desviació soferta durant el projecte	63
6.3.1 Canvis respecte les tasques	63
6.3.1.1. Familiarització amb OPC-UA	63
6.3.1.2. Adaptació a les característiques de NECADA	63
6.3.2 Canvis de termini	64
6.3.3 Proves amb iOS	64
6.4. Planificació final	65
6.4.1 Càrrega de treball de les tasques	65
6.4.2 Diagrama de GANTT	65
7. Gestió econòmica	67
7.1. Costos directes	67
7.1.1. Costos en hardware	67
7.1.2. Costos en software	68
7.1.3. Costos en recursos humans	68
7.1.4. Costos directes totals	69

7.2. Costos indirectes	69
7.2.1. Electricitat	70
7.2.2. Internet	70
7.2.3. Costos indirectes totals	70
7.3. Costos directes i indirectes	71
7.4. Costos de contingència	71
7.5. Costos d'imprevistos	71
7.6. Costos totals i control de gestió	72
8. Identificació de lleis i regulacions	73
9. Informe de Sostenibilitat	75
9.1. Dimensió econòmica	75
9.2. Dimensió ambiental	75
9.3. Dimensió social	76
10. Conclusions del projecte	77
11. Referències	78

Índex de taules

Taula 1: Cas d'ús "Iniciar sessió"	30
Taula 2: Cas d'ús "Tancar sessió"	31
Taula 3: Cas d'ús "Consultar el llistat de factors"	31
Taula 4: Cas d'ús "Consultar l'estat general dels factors"	32
Taula 5: Cas d'ús "Consultar les dades específiques de cada factor"	32
Taula 6: Cas d'ús "Actualitzar informació dels factors"	33
Taula 7: Cas d'ús "Consultar estat general d'estalvi del sistema"	33
Taula 8: Cas d'ús "Consultar estat actual de la situació"	34
Taula 9: Cas d'ús "Consultar el perfil de l'usuari"	34
Taula 10: Cas d'ús "Consultar informació sobre l'aplicació"	35
Taula 11: Requisit Volere "10.a Appearance"	35
Taula 12: Requisit Volere "11.a Ease of use"	36
Taula 13: Requisit Volere "11.c Learning"	36
Taula 14: Requisit Volere "12.d Reliability and Availability"	36
Taula 15: Requisit Volere "12.e Robustness or FaultTolerance"	37
Taula 16: Requisit Volere "12.g Scalability or Extensibility"	37
Taula 17: Requisit Volere "14.c Adaptability"	38
Taula 18: Requisit Volere "15.c Privacy"	38
Taula 19: Total d'hores del treball per tasques inicial	61
Taula 20: Total d'hores del treball per tasques final	65
Taula 21: Costos directes (Hardware)	67
Taula 22: Costos directes (Software)	68
Taula 23: Costos directes inicials (Recursos humans)	68
Taula 24: Costos directes finals (Recursos humans)	69

Taula 25: Costos directes	69
Taula 26: Costos indirectes	70
Taula 27: Sumatori de costos directes i indirectes	71
Taula 28: Costos de contingència	71
Taula 29: Costos d'imprevistos	72
Taula 30: Cost total amb IVA i sense	72

Índex de figures

Figura 1: Resum de com funciona Xamarin	23
Figura 2: Exemple d'Internet sobre el render de Xamarin	24
Figura 3: El patró MVVM	25
Figura 4: Diagrama de casos d'ús de l'aplicació	30
Figura 5: Diagrama de classes UML de l'aplicació	39
Figura 6: Fitxers principals que componen l'aplicació	41
Figura 7: Arquitectura del sistema de la solució prototipada	42
Figura 8: Vista principal de la versió Android	44
Figura 9: Vista principal de la versió iOS	44
Figura 10: Vista d'inici de sessió de la versió Android	45
Figura 11: Vista d'inici de sessió de la versió iOS	45
Figura 12: Vista del detall del factor de la versió Android	46
Figura 13: Vista del detall del factor de la versió iOS	46
Figura 14: Vista del detall de la situació de la versió Android	47
Figura 15: Vista de detall de la situació de la versió iOS	47
Figura 16: Vista del perfil del factor de la versió Android	48
Figura 17: Vista del perfil del factor de la versió iOS	48
Figura 18: Vista de la informació de l'aplicació de la versió Android	49
Figura 19: Vista de la informació de l'aplicació de la versió iOS	49
Figura 20: Vista del splash de la versió Android	50
Figura 21: Vista del splash de la versió iOS	50
Figura 22: Paleta dels colors verds utilitzats a l'aplicació	51
Figura 23: Paleta dels colors taronges utilitzats a l'aplicació	51
Figura 24: Paleta dels colors vermells utilitzats a l'aplicació	51

Figura 25: Paleta dels colors blaus grisosos utilitzats a l'aplicació	52
Figura 26: Icones utilitzades pels dotze factors a l'aplicació	53
Figura 27: Logotip de l'aplicació amb fons i sense	53
Figura 28: Diagrama de navegabilitat de l'aplicació	54
Figura 29: Un dels hotels que conforma el hotels.json	60
Figura 30: Diagrama de Gantt inicial	62
Figura 31: Diagrama de Gantt final	66

Glossari

Les paraules que es troben aquí explicades estan marcades en negreta al llarg de la memòria, exceptuant les paraules que formen part de les taules o els peus de taula/figura.

- **Assemblies:** Són blocs de construcció de codi de .NET. Una assembly és una col·lecció de tipus i recursos que formen una unitat lògica i estan construïts i han de treballar conjuntament.
- **Code-behind:** Són els fitxers de C# que estan associats a cada fitxer XAML i que per defecte es necessiten una a l'altra per funcionar correctament.
- **colorAccent:** És la variable que utilitza Android per definir el color d'alguns elements de la interfície d'usuari com checkboxes, radio buttons, caixes de text editable...
- **Debug:** És un tipus d'execució de programa que permet disposar d'una eina de resolució de problemes podent analitzar com funciona el codi i detectar els errors d'aquest.
- **Encoding:** És el procés que converteix informació d'un determinat origen en símbols per la comunicació o l'emmagatzemament de dades.
- **Flat Design:** El disseny on qualsevol cosa que no aporta valor al missatge que es vol transmetre no és necessari.
- **GitHub:** S'utilitza bàsicament per la generació de codi de programes. Permet allotjar projectes utilitzant el sistema de control de versions Git.
- **GUI:** La interfície gràfica d'usuari (GUI) és la part visual dels sistemes informàtics amb que estem acostumats a tractar els humans.
- **Harmonia complementària:** És un tipus d'harmonia que s'aconsegueix quan es combinen dos colors oposats a la roda de colors. És una harmonia que intrínsecament és de contrastos.
- **Harmonia d'anàlegs:** És un tipus d'harmonia que s'obté en una composició en què s'utilitzen tons que es troben pròxims en el cercle cromàtic.
- **IDE:** Un l'entorn integrat de desenvolupament és una eina informàtica que permet generar codi de manera còmode i ràpida.
- **IOT:** El Internet de les coses (Internet Of Things) es refereix a una xarxa d'objectes cotidians interconnectats.
- **Target version:** És la versió de sistema operatiu a que va dirigida una aplicació Android. Aquesta definició ajuda a que a l'hora de l'execució ho faci en unes condicions més òptimes per la versió especificada.

- **Teamfoundation:** És una eina de Microsoft que permet gestionar el codi amb un control de versions propi o amb Git.
- **XML:** És un metallenguatge extensible d'etiquetes. És un estàndard per l'intercanvi d'informació estructurada entre diferents plataformes.

1. Introducció i contextualització

1.1. Context

1.1.1. Introducció

Una de les principals preocupacions de la humanitat en aquests últims anys és transformar el món perquè sigui sostenible i que permeti seguir-hi vivint durant molts més segles. A dia d'avui malgastem més del doble de l'energia que utilitzem cada any [3], cosa que provoca uns efectes molt nocius al planeta terra, i la informàtica hi té un pes molt significatiu en aquesta deterioració [4][5]. No obstant això, cada vegada apareixen més projectes que intenten capgirar aquesta tendència i des de la informàtica aportar a una visió més responsable sobre el malbaratament i la contaminació.

L'objectiu d'aquest projecte és dissenyar una aplicació multiplataforma per a l'estalvi energètic en el sector hotelier. L'eina que s'ha utilitzat per trobar els valors òptims reduint la demanda d'energia o consum d'aquests edificis ha estat NECADA [6].

NECADA, que prové de “No Emissions CAD for Architecture”, és una infraestructura que permet la simulació del consum d'un edifici i trobar-ne el valor òptim que redueixi l'energia consumida de l'edifici [6]. Això permet obtenir un bon grapat de dades que per exemple es poden utilitzar per reduir el malbaratament.

1.1.2. Formulació del problema

Actualment un dels problemes més grans de malbaratament d'energia és la que gasten els edificis, ja que podria ser molt optimitzada. Aquests suposen un 40% del consum d'energia total en el cas d'Espanya [7] i en uns percentatges molt similars a d'altres països avançats com poden ser Estats Units d'Amèrica o el Regne Unit [8].

Per tant, el problema principal és que s'efectua un gran consum d'energia i es genera una quantitat enorme d'emissions de CO₂ que es podrien evitar. La manera com es podria esquivar una gran part d'aquesta despesa d'energia d'edificis seria sent-ne conscients de l'energia que estem gastant en cada moment, i sense afectar de cap manera a la nostra salut ni a la qualitat de vida, intentar minimitzar els usos excessius de diferents aparells o sistemes. És molt difícil pretendre que una persona es faci càrrec de totes aquestes qüestions, i a més a més ho faci d'una forma encertada i és per això que es poden desenvolupar softwares que facin un monitoratge i estudi d'aquestes despeses i s'encarreguin de l'optimització pertinent de l'energia. El programari que s'encarrega de dur a terme tasques d'aquest tipus s'anomenen amb l'acrònim de EMS i cadascun té en compte uns factors i estàndards diferents per crear aquest model d'optimització.

Així doncs, amb el programari de gestió d'energia NECADA com a punt de partida, es va tenir com a objectiu del projecte: dissenyar una aplicació multiplataforma per a l'estalvi energètic

en el sector hotelier. Amb aquest sistema es podia fer tota la recollida de dades necessàries i a partir d'aquí calia estudiar i mirar com presentar-les perquè l'usuari es veiés encara més predisposat a estalviar energia.

1.2. Estat de l'art

En aquesta secció es pretén donar informació més a fons sobre els diferents temes dels quals s'ocupa el projecte. Així doncs, en el següents apartats s'analitza un per un els diferents apartats que componen el treball.

1.2.1. L'estalvi energètic en la informàtica

L'estudi que s'encarrega de l'estalvi energètic en la informàtica i que sigui responsable i sostenible, és el Green Computing. Aquesta es basa en diferents tipus de pràctiques com poden ser la gestió de l'energia, l'ús de materials reciclats, virtualització, o aplicacions que intenten canviar els hàbits de les persones amb l'objectiu de conservar energia [9][10].

La primera aparició del moviment del Green Computing va tenir lloc l'any 1992 quan la agència de protecció ambiental del govern d'Estats Units va crear el Energy Star. Aquest programa consistia en l'etiquetatge de diferents tipus d'aparells electrònics segons la seva eficiència energètica, la seva intenció era minimitzar l'ús d'energia, maximitzar l'eficiència de l'aparell i finalment promocionar aquest tipus d'aparells [11][12].

Una de les primeres fites que va aconseguir el Energy Star va ser la introducció en les pantalles d'ordinador del Sleep mode, que consisteix a posar l'aparell en un mode de pausa quan no detecta activitat de l'usuari [13].

A mesura que van anar passant els anys cada cop aquests ideals es van anar estenent més i més pel planeta i va acabar suposant que moltes de les directives acabessin en lleis per protegir el medi ambient [9]. A dia d'avui per seguir tenint o obtenir una etiqueta del precursor Energy Star, hi ha un control encara més exhaustiu de les característiques del producte i es tenen en compte encara més factors ambientals [10].

1.2.2. Estudis de EMS

Avui en dia existeixen molts tipus de EMS però tots tenen una cosa en comú: cadascun s'encarrega de controlar la despesa d'energia i utilitzar aquesta informació per a que l'usuari (o l'automatització) pugui gestionar-ho correctament, o pugui conscienciar-se'n.

Aquests sistemes de gestió d'energia (EMS) van aparèixer aproximadament el 2009, quan les grans corporacions s'hi van posar en marxa. La idea principal era que tinguessin un ús domèstic i se'n van treure prototips que van acabar sense èxit i els projectes van desaparèixer [13].

A dia d'avui existeixen moltes solucions diferents per a poder estalviar energia. Alguns exemples [14] d'aquestes serien: ACE [15], Meter Readings [16], Wegowise [17] o NECADA

[6]... Tots aquests utilitzen la tecnologia per intentar que els usuaris facin un bon ús de l'energia buscant apropar-los a uns valors òptims. Alguns d'aquests es poden complementar per crear nous sistemes intentant aconseguir un estalvi major aprofitant les dades recollides. Com podem veure a la Taula 3 de la referència 12, existeixen moltes opcions diferents però poques es queden amb el millor de cada una o realment intenten connectar-se amb altres sistemes.

1.2.3. Les aplicacions gamificades

Aquest tipus d'aplicacions han sigut tendència aquests últims anys i ho podem veure clarament ja que molts mitjans de comunicació ja fa temps que se'n fan ressò.

La gamificació és fer viure una experiència com si fos un joc quan realment no ho és [18]. La finalitat és motivar les persones a aprendre, influir i compartir coneixement, entre d'altres valors positius que aporten els jocs [19][20].

Hi ha qui vol vendre aquesta estratègia com una cosa molt nova, però aquesta potència que pot tenir el joc ha estat manifesta per part de molts professors i formadors que porten temps aplicant-ho. De fet hi ha estudis des de fa molt de temps que en confirmen la seva utilitat [21][22].

L'experimentalització en aplicacions s'aplica gairebé des de l'inici de l'aparició d'aquestes però no s'ha vist com un mètode per captar l'atenció de l'usuari i com una gran oportunitat per motivar-los fins uns anys després, cap al 2008. De fet cada cop estem més envoltats d'elements pensats des de la gamificació i ni ho tenim en compte i a més a més afecten i produeixen aquestes ganes de completar el que es demana [23].

D'altra banda també hi ha un problema a la indústria de les aplicacions: estudis recents confirmen que un 21% d'aplicacions mòbil s'utilitzen només un cop i ja mai més se'n torna a fer ús. Un dels problemes principals d'això és que aquestes no enganxen suficientment als seus usuaris que finalment opten per altres solucions o per simplement desfer-se'n [24].

Pel que es refereix al tipus d'aplicació gamificada que es vol aplicar, existeixen moltes solucions que t'ajuden a ser més considerat amb el medi ambient i tenir-ne cura, fins i tot hi ha qui parla de Green Gamification: N'hi ha que et faciliten donar-te de baixa d'empreses que t'envien correu brossa a la bústia cada setmana, també n'hi ha que t'encoratgen a fer com a mínim una acció bona pel planeta al dia i fins i tot n'hi ha que t'ajuden a canviar hàbits del dia a dia per estalviar energia [25].

2. Abast

2.1. Objectius

La despesa innecessària d'energia es genera a gairebé la totalitat dels edificis i per tant cal un canvi de paradigma per no malgastar tota aquesta energia que es consumeix (i per tant es paga) i a més a més contamina greument el nostre entorn.

La globalització ha contribuït a que cada cop més persones viatgin i visitin altres societats, pobles i cultures, i la gran majoria d'aquestes s'allotgin en hotels. Tal i com comentàvem, quasi tots els edificis malgasten energia per si mateixos ja que el personal que els gestiona no sap com optimitzar la despesa, però és que a més a més s'hi afegeix un factor rellevant que ajuda poc a que l'usuari que s'hi allotja en faci un bon ús. Un client d'un hotel en la major part del casos paga un preu fix d'un bon inici independentment de l'ús que en faci del seu allotjament (si és que no el fa malbé) i el preu només va en funció del servei que se li ofereix. Així doncs, l'usuari no rep cap mena de benefici (econòmic) si procura fer-ne un bon ús i per exemple li suposa el mateix obrir l'aire condicionat a totes hores que no fer-ho, i si aquesta persona no està conscienciada sobre el problema que això suposa li pot semblar completament igual la despesa energètica de la seva habitació.

Per tant, s'ha de treballar en la direcció de que un usuari no només ha de pagar pel servei que se li ofereix, sinó que ha de poder sortir-ne recompensat en cas d'utilitzar-ho correctament. I això no ha de suposar una despesa per l'hotel, sinó que aquest tindrà un estalvi significatiu (en cas que els clients ho utilitzin) que llavors una part de l'estalvi econòmic podrà ser retornada al consumidor pel seu bon ús.

A més a més, l'usuari no haurà de calcular ni imaginar-se si realment està fent el millor per l'hotel adherit a aquest sistema, sinó que l'aplicació multiplataforma creada li permetrà saber-ho en tot moment i de forma fàcil i atractiva. Per si no fos poc, s'incentivarà l'ús d'aquesta aplicació amb diferents tècniques de gamificació (com per exemple el sistema de retorn de diners que s'ha comentat). Com s'ha vist ja existeixen aplicacions que intenten canviar els hàbits ambientals de l'usuari mitjançant el joc, però aquest és un projecte a part que permet obtenir la informació de la situació d'una habitació d'un hotel, que proporcionarà la eina de gestió de energia, i permetrà saber explícitament com estalviar energia.

El principal objectiu doncs, és mitjançant l'eina d'anàlisi (simulació) NECADA que aportarà la informació necessària analitzant aquestes dades i mostrant-les adequadament a l'usuari, tan per Android com iOS, tot incentivant l'ús de l'aplicació de forma còmode.

2.2. Abast

2.2.1. Abast inicial

Per poder complir tots els objectius que es busquen en aquest projecte s'havia de crear una aplicació amb tots els matisos definits a l'Estat de l'art com a eixos de treball.

A dia d'avui NECADA té una sèrie de funcionalitats que seran les que posaran límits a les qualitats que es poden analitzar de l'edifici i per tant optimitzar. Les funcionalitats que es van utilitzar van ser les que formaven part de l'exemple de dades de prova de NECADA, amb el qual s'ha treballat, i per tant en un cas futur de disposar més dades podrien aparèixer més factors.

Pel que fa a tècniques de gamificació, es van analitzar les tècniques més rendibles i utilitzades per a aquest tipus d'estratègia tenint en compte que sigui compatible amb la seva usabilitat. Per tant, aquesta aplicació haurà de ser fàcilment usable: sobretot eficaç en el seu ús i de fàcil aprenentatge. S'havia de considerar que seria utilitzada per usuaris amb una gran diferència d'edat, i per tant ha de poder ser apte per tothom. A més a més, ha de ser atractiva per els usuaris, aquests han de tenir una experiència agradable en el seu ús i se'ls hi ha de mostrar de manera fàcil i intuïtiva.

2.2.2. Abast final

L'abast de l'aplicació es el mateix que està definit en l'apartat anterior, però en algun aspecte va canviar una mica respecte on es pretenia arribar d'un bon començament:

L' OPC-UA, que és un estàndard de IOT, permet la comunicació entre sensors de manera unificada, per poder rebre'n informació, i es pretenia saber-lo utilitzar per poder comunicar-se amb la situació real d'un edifici. Es va veure que era quelcom molt costós i sortia realment del què es volia treballar en aquest projecte i es va decidir treballar en una versió de proves local i que conseqüentment s'adaptessin certes dades que no podrien obtenir-se al no treballar amb un sistema connectat amb uns sensors que recullen dades a temps real.

En un primer moment també es volia ser més ambiciós amb la gamificació del projecte, però també es va veure que es podien incloure idees de com funcionaria aquesta aplicació un cop al mercat. Finalment però poques idees de gamificació es podrien implementar a l'aplicació si realment en el que es volia centrar era en com mostrar amb les dades, dissenyar-la multiplataforma nativament i com dissenyar-ho i fer-ho atractiu per a l'usuari.

Així doncs, es pot dir que l'OPC-UA va quedar totalment fora d'abast i la gamificació del projecte si que es va incloure perquè està pensada per ser aplicada en molts aspectes fent-la atractiva com si fos un joc però sense tanta ambició com hi havia d'un bon principi. La presentació dels diferents factors a l'aplicació amb els colors que n'indiquen l'estalvi, proporciona una part de gamificació, i també que en cas d'arribar a un cert percentatge d'estalvi el sistema gratifiqui a l'usuari pel bon ús. Hauria estat bé poder ser més ambiciós al respecte, però el projecte realment es centrava més en altres aspectes. Pel que fa al retorn

monetari dependent de l'estalvi tampoc es va poder implementar ja que no hi havia manera de poder quantificar aquestes dades amb la informació que proporcionava la versió de proves.

Ha de quedar clar que aquest projecte és d'un prototip de la idea futura amb la que es va treballar. Les dades que s'utilitzen de NECADA són d'un projecte de prova i sense treballar a temps real amb la sensòrica de cap edifici, la idea però és que l'aplicació està preparada i té tota l'estructura per treballar amb les dades reals tal i com les emmagatzema el sistema de gestió d'energia. La connexió amb un sistema a temps real hauria afegit un plus que es va considerar que no tenia a veure amb el que realment es volia cercar en el treball. Per tant al treballar amb un prototip, com per exemple la base de dades d'hotels disponibles al sistema, es simularen per mostrar com hauria de ser un cop implementada, es detallarà més endavant com es van definir.

2.3. Metodologia i rigor

Per a poder aplicar la metodologia pertinent amb el rigor necessari primer de tot va caldre fer una bona planificació dels mesos (com es veu en el punt 6) que hi havia per davant per poder sospesar correctament la càrrega de treball i que el resultat llest. Per poder arribar a aquest punt, primer calia definir com es va desenvolupar aquest projecte, i com es va seguir i validar.

2.3.1. Eines de desenvolupament

A mesura que s'avançava s'havia de mostrar una relese que funcionés al director, i per tant va semblar que fer una gestió àgil del projecte era el que més interessava, concretament de Scrum. Tot i això, aquest tipus de metodologia està molt orientada per processos en grups però finalment es tracta treure'n la part aplicable i que pot funcionar. Així que bàsicament es va desglossar el projecte en petites parts per anar fent iteracions setmana a setmana i en acabat arribar còmodament al resultat final. Com que podien aparèixer diversos factors o decisions que podien acabar alterant certs requeriments, Scrum permetia ser més flexible als canvis.

Per al correcte desenvolupament de l'aplicació i tenir disponibles totes les versions que s'havien anat generant, es va utilitzar **Teamfoundation** que va servir com a control de versions. Al final també va acabar produint una bona documentació de tots els canvis que es van anar aplicant i finalment en va acabar sortint un backlog.

2.3.2. Eines de seguiment

Per efectuar el seguiment telemàtic es va utilitzar One Drive per a revisar la documentació, ja que aquest permetia treballar amb documents Word sense possibles errors de conversió de fitxers com algun cop havia passat al principi utilitzant Google Drive. Per revisar el codi i veure com s'avançava en el projecte també teníem: **Teamfoundation**.

Per a la comunicació més directe amb el director es va utilitzar el Skype. En cas de sorgir problemes més importants i de no poder-se solucionar de cap altra manera, es va concretar visites presencials però no era la idea inicial.

El seguiment del projecte es podia efectuar bàsicament des de **Teamfoundation**, ja que aquest permetia al director del projecte poder-lo supervisar en tot moment, també podia fer-hi comentaris sobre els aspectes que creia més rellevants. Fins i tot el director, a través de **Teamfoundation**, podia compilar i provar les releases que s'anaven generant.

El que s'ha comentat anteriorment és la idea principal de com es podia fer el seguiment, tot i que com es va funcionar bàsicament va ser fent trucades setmanals per Skype, on es compartia la pantalla amb el director i se li ensenyava els avenços de les releases del sprint que s'havia fet. No totes les setmanes hi va haver trucades, per exemple quan no hi havia cap mena de release, però a vegades també s'esperava ajuntar uns quants sprints fins que no se'n feia una. Es va fer d'aquesta manera ja que era com anava millor a les parts implicades per disponibilitat horària i compartir les opinions ja que així era instantani el feedback.

2.3.3. Mètode de validació

A diferència d'altres metodologies, en les àgils mentre es fan les proves del codi, de manera paral·lela s'anava fent el desenvolupament de l'aplicació. Així doncs, els test es feien en acabar cada sprint un cop s'havia treballat en totes les noves funcionalitats.

Per tant cada release s'havia validat la part de projecte programada i es va fer mitjançant les eines de Visual Studio, com per exemple l'emulador d'Android. On van aparèixer problemes per poder provar que el codi funcionava correctament era per la versió de iOS, ja que Apple no permet que es pugui provar el codi de cap manera si no es té un Mac. N'explico la problemàtica més detalladament a l'apartat 6.3.3. d'aquest treball. Tot i la problemàtica finalment es van poder provar els simuladors d'Apple i testejar que tot funcionés correctament.

Els simuladors que es van utilitzar per a cada sistema operatiu van ser escollits en funció del sistema operatiu que portaven incorporat i la mida de la seva pantalla perquè les comparacions que es fessin entre els dos dissenys fossin el màxim de similars. Els escollits van ser el Nexus 5 i l'iPhone 6 Plus. Més endavant en el punt 3.6. s'explica concretament quins van ser els motius d'escollir aquests dos dispositius.

Per altra banda, si el director veia que el projecte no s'encaminava correctament o podia es podia millorar, ho comprovava i llavors es passava a introduir les noves tasques a la següent iteració.

3. Tecnologies, eines, conceptes i coneixements aplicats

3.1. Llenguatges de programació

Per a la creació de l'aplicació multiplataforma es va haver d'aplicar diferents llenguatges que mai abans s'havien utilitzat: com és el cas del C# i el XAML, que són els dos llenguatges principals que s'utilitzen pel desenvolupament d'aplicacions Xamarin. En aquest apartat s'analitzen les principals característiques d'aquests llenguatges:

3.1.1. C#

És un llenguatge de programació orientat a objectes que va ser desenvolupat per Microsoft i està emmarcat dins de la plataforma .NET que és l'entorn de creació, distribució i execució de les aplicacions de Windows.

Aquest llenguatge prové del C++ i és per això que no va ser difícil familiaritzar-s'hi, ja que el C++ va ser el primer llenguatge que es va aprendre a la carrera. Pel desenvolupament de l'aplicació s'utilitza bàsicament aquest llenguatge, i per tant, entendre'l bé va ser clau per la creació de l'aplicació.

3.1.2. XAML

És un llenguatge de marques que s'utilitza per la definició d'interfícies d'usuari. És una variant que Windows es va inventar sobretot per fer més fàcil el disseny de la GUI, anteriorment la interfície s'editava a partir del C# mateix, i la idea de separar-ho entre el XAML i el codi C# associat a la vista va permetre a dissenyadors i desenvolupadors poder treballar per separat i també fer-los-hi la feina més fàcil als dissenyadors d'interfícies [26].

Fins i tot ja existeixen dissenyadors visuals de interfícies que estalvien escriure el codi en XAML, tot i que no s'ha utilitzat en el projecte [27].

3.2. Visual Studio

És l'IDE de Microsoft el que s'utilitza per a generar aplicacions per Windows, Linux o Mac. Aquest programari va permetre crear l'aplicació Xamarin sense cap mena de problema. Té integrades moltes eines com poden ser el gestor de paquets NuGet, el control de versions TeamFoundation o els simuladors.

La versió que es va utilitzar era la 2017 ja que era la més nova i estable en el moment de començar el projecte.

3.3. Xamarin.Forms

Xamarin presenta un seguit d'eines per generar aplicacions multiplataforma de forma nativa per Android, iOS i Universal Windows Platform (tot i que només es van utilitzar les dues primeres) utilitzant únicament C# a través de Visual Studio. És propietat de Microsoft i està distribuït de forma gratuïta.

Quan es parla de forma nativa es refereix a que està desenvolupada i optimitzada específicament per un sistema operatiu concret i la plataforma de desenvolupament del fabricant. Aquestes aplicacions s'adapten completament amb les funcionalitats i característiques del dispositiu, resultant una millor experiència d'usuari [28][29].

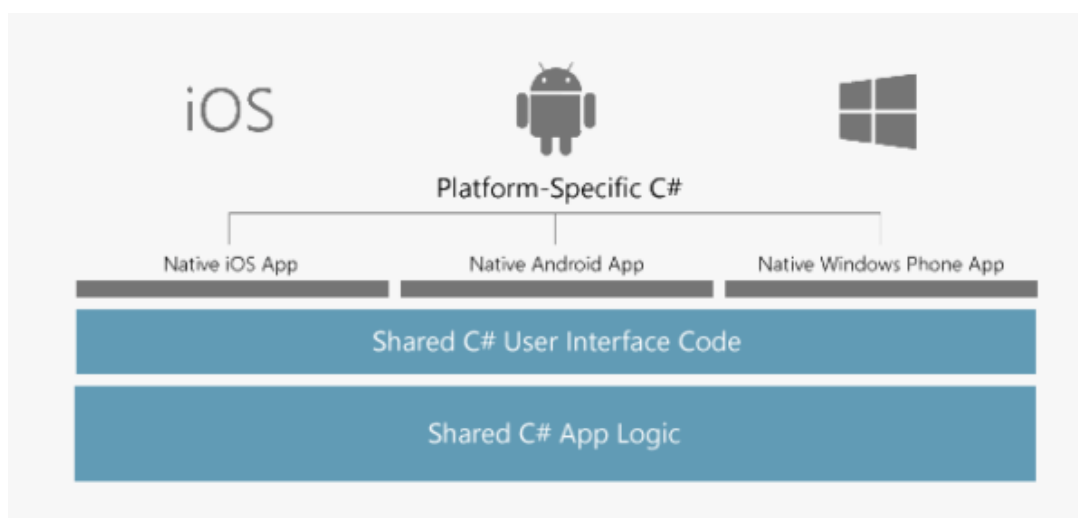


Figura 1: Resum de com funciona Xamarin

El que permet doncs és quelcom molt potent: a partir d'un codi, programat en C# i parts de XAML, crear una aplicació per diferents sistemes operatius de manera nativa, deixant enrere el sistema habitual que és que cada sistema operatiu es desenvolupi amb les seves eines i llenguatges específics.

Anem a veure com funciona el codi compartit a través d'un exemple:

El codi que es programa majoritàriament forma part d'un codi compartit entre plataformes que és utilitzat per aquests sistemes operatius.

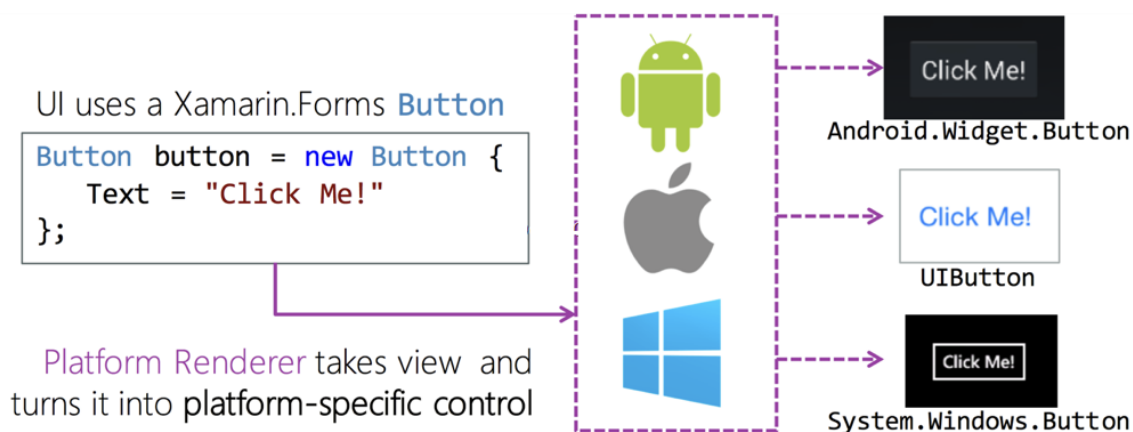


Figura 2:

Exemple d'Internet sobre el render de Xamarin [30].

Podem veure doncs que Xamarin s'encarrega de convertir les expressions programades al codi compartit a l'equivalent de cada sistema operatiu.

Un dels problemes principals de Xamarin però és que tot i tenir una bona documentació, a vegades es fa difícil que sorgeixin errors i es poden trobar pocs llocs on solucionar els dubtes. Encara hi ha poca comunitat que participi als fòrums i resolgui dubtes relacionats amb el sistema de desenvolupament, tot i que la gent que hi participa són persones molt actives. De fet més d'un cop es va participar en un fòrum de dubtes i la comunitat va ajudar i va respondre. El problema és que quan hi ha llançaments relativament nous hi ha poca informació disponible, i moltes vegades es pot tornar una experiència feixuga. Per altra banda, la mida de les aplicacions poden tenir un pes el 30% més gran que les mateixes aplicacions creades amb l'IDE i codi original de cada plataforma, això és degut a les **assemblies** que necessita l'aplicació per poder executar-la nativament [27]. El pes podria suposar un problema pels futurs usuaris però realment no serien tants MB com per impedir-li a l'usuari baixar-se l'aplicació, així que no es tindrà en compte.

També és interessant tenir la mirada posada en el futur, ja que el 2019 l'equip de Microsoft ha anunciat que amb la nova versió de .NET (la 3.0) hi haurà possibilitat de construir projectes també per Web, per tant, potser s'està a prop d'aplicacions multiplataformes natives i també amb versió web [31].

3.3.1. Embedded Resource

Un Embedded Resource és un recurs que s'emmagatzema amb l'aplicació, però en comptes de tenir una còpia a cada plataforma diferent senzillament guarda el fitxer a les carpetes compartides entre sistemes operatius [32] i es accessible des de cada sistema.

Al llarg del projecte es va utilitzar molt aquest sistema per carregar tots els icones dels factors, les imatges de detall o els fitxers locals necessaris per l'aplicació com les dades de prova de NECADA.

3.4. MVVM

El patró general que es va utilitzar en aquesta aplicació va ser el MVVM (Model - Vista - Vista Model). Aquest permetia separar amb molt poques dependències la interfície d'usuari i tota la lògica de l'aplicació, permetent el treball simultani de dissenyadors i desenvolupadors, facilitant el testing, el manteniment i la reutilització del codi.

Model:

El Model es refereix al domini del projecte, tota la capa de dades i d'informació amb que es treballa.

Vista:

La Vista és la part més coneguda del sistema ja que és la part amb la l'usuari interactua, sent la interfície de l'aplicació, és a dir la presentació.

Vista Model:

La Vista Model implementa comandaments i conté les propietats que la Vista necessita utilitzar i incorporar. Aquesta notifica a aquesta de les modificacions o canvis especificats. Així doncs, la funció principal és integrar les funcionalitats que s'han d'oferir a l'usuari mitjançant la Vista [33].

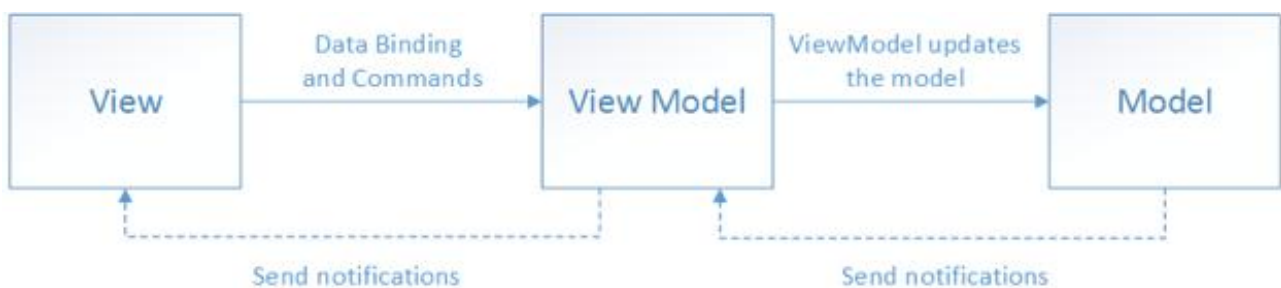


Figura 3: El patró MVVM [34].

Cada una d'aquestes capes queden totalment diferenciades per carpetes en el codi font de l'aplicació i no hi ha d'haver cap problema per identificar-les.

3.5. Llibreries de NuGet

NuGet és un gestor de paquets gratuït i de codi obert creat per Microsoft. És el gran repositori on rauen totes les llibreries per a .NET i conseqüentment també per Xamarin. Permet crear, compartir i consumir llibreries als desenvolupadors [35][36].

En aquest projecte s'han utilitzat diverses llibreries que s'han aconseguit a través de NuGet, són les següents:

3.5.1. FlowListView

Aquesta llibreria va permetre implementar llistats (ListView) també horitzontals, és a dir per columnes. De bon principi es va pensar en un disseny de llistat de blocs dividits en dues columnes i aquesta implementació de Daniel Luberda va permetre poder dissenyar l'aplicació tal i com s'havia pensat.

En la seva implementació hi va haver alguns problemes, ja que al ser una llibreria molt concreta a vegades costa trobar molta informació tot i que els creadors solen documentar-ho correctament. Més d'una vegada es van resoldre els dubtes a través de fils de discussió de **GitHub**. Se'n deixarà la referència a continuació [38].

3.5.2. PullToRefresh

Aquesta llibreria va permetre afegir un Layout que integrés l'acció de refrescar la Vista i així poder executar qualsevol funció necessària. Aquesta acció es va integrar a la part superior de la pàgina principal per poder renovar les dades dels factors [39].

3.5.3. Settings

Aquesta llibreria permetia accedir a la configuració del dispositiu i emmagatzemar privadament dades clau-valor. Va servir per emmagatzemar dades molt simples i no haver d'implementar una base de dades que hauria estat més costosa i no era necessari [40][41].

3.5.1. JSON.NET

Aquesta llibreria és un framework per .NET que va permetre deserialitzar les dades de diferents arxius JSON als models que es necessitava [42].

Hi ha moltes metodologies diferents per deserialitzar fitxers JSON però es va creure que la millor manera era utilitzar JSON.NET ja que a diferència d'altres ho permetia fer directament, simplement mantenint els noms i estructura del model i el JSON en qüestió.

3.6. Android i iOS

En el moment de començar a desenvolupar l'aplicació calia decidir en quina seria l'API mínima que permetria córrer l'aplicació per Android i també ajustar la versió mínima suportada per iOS. La versió mínima per iOS es va escollir la que estava per defecte, la 8.0, ja que més del 98% [43] dels dispositius amb aquest sistema operatiu tenen aquesta versió com a mínim. Pel que fa la versió mínima de Android es va decidir la API 19 (KitKat) que és una API que no és molt vella (del 2014) i més del 95% [44] dels dispositius actuals tenen aquesta o versions més noves, per tant era òptima per ser la mínim a la que se li donava suport [45]. Escollir aquesta versió mínima era important, ja que l'aplicació podrà dur a terme certes funcions que com a mínim pugui fer la versió escollida, sinó certes funcionalitats quedaran fora i s'hauran d'utilitzar metodologies antigues per programar, i és per això que era important saber-la triar bé per aconseguir que un gran ventall de població

pogués tenir disponible l'aplicació al mòbil i per altra banda no fer perdre facultats a l'aplicació. Com que tampoc integra funcionalitats molt avançades tampoc va suposar gaires problemes a l'hora de veure com d'antiga es podia agafar l'aplicació.

Per Android també es va haver d'escollir el **target version**, i la que es va escollir va ser la Oreo ja que és una de les últimes que va sortir i cada vegada més usuaris hauran d'anar millorant les versions dels seus dispositius i aquesta és una aplicació que mira de cara al futur i per tant haurà de funcionar bé en els dispositius que tindran els usuaris del futur.

Els dispositius que s'havien d'utilitzar per fer les simulacions de l'aplicació, només calia que tinguessin una versió igual o més alta que la que s'havia triat com a mínima, i en el cas del Nexus 5 (5") i iPhone 6 Plus (5.5") tots dos dispositius superaven les versions mínimes corresponents i a més a més la mida de les seves pantalles era molt similar cosa que permetia veure més clarament com quedava l'aplicació en cada sistema operatiu.

4. Anàlisi de requisits i especificació

4.1. Stakeholders

Degut al gran interès que tenen les persones o empreses que han de pagar l'energia per reduir-ne costos, es tracta d'un projecte interessant que les part a qui anirà dirigit en treuen un benefici molt gran: econòmic, social i ambiental, que provindrà d'optimitzar una energia que ara mateix s'està malgastant i en canvi podria ser estalviada [46].

Els sistemes de gestió d'energia necessiten un modelatge dels edificis que analitzen per poder fer tots els càlculs necessaris per saber com optimitzar l'energia, així que sent conscient d'això es va considerar oportú centrar-se en un tipus d'edifici, com són els hotels, ja que és un espai on l'empresa i el client en poden treure benefici a partir de l'estalvi d'energia i per tant són dues parts potencialment interessades.

Aquesta aplicació té repercussió sobre tot el planeta ja que una de les coses que busca és reduir la contaminació d'aquest, però es dirigeix específicament a uns certs actors:

4.1.1. Sector hotelier

El sector de l'hostaleria és un dels màxims beneficiats en aquest projecte ja que el seu treball per poder integrar el sistema dins del circuit no és gaire costós però en canvi els guanys que s'obtindran gràcies a l'estalvi de l'energia són molt importants: energètics i econòmics.

4.1.2. Els clients dels hotels

La idea principal que regeix l'aplicació és que també aporta un benefici per als clients dels hotels, incentivant així el seu ús. Aquests usuaris tindrien un guany econòmic si gestionen correctament l'estalvi de l'habitació de l'hotel en que es troben: si quan no hi ha ningú a una habitació apaguen la llum, si no es deixen les aixetes obertes més del compte... NECADA capturaria aquestes dades i l'aplicació ho gestiona fent que l'usuari pugui veure si ho està fent correctament.

4.1.3. Departament de Territori i Sostenibilitat

El Departament de Territori i Sostenibilitat de la Generalitat de Catalunya s'encarrega de la gestió pública del medi ambient i la sostenibilitat i per tant, serà una part interessada en un projecte d'aquestes característiques que mira d'aportar una correcta gestió a nivell social i ambiental del malbaratament de l'energia i emissions de CO₂ dels edificis. Per molt que sigui una iniciativa privada, els objectius finals de l'aplicació coincideixen amb els del departament i és per això que també és una part interessada del projecte.

4.2. Requisits funcionals

Tot projecte software té un seguit de requisits funcionals que defineixen tots els usos que se'n pot fer de la plataforma. Seguidament es llisten aquests requisits amb les explicacions de la funcionalitat que permetrà:

1. Iniciar sessió: permet iniciar sessió a l'aplicació a aquells usuaris que coneguin les seves credencials i aquestes coincideixin amb les que conté l'arxiu que l'hotel els hi haurà lliurat prèviament. Permetrà iniciar la sessió entre els dies d'estada a l'hotel (que la clau conté).
2. Tancar sessió: permet tancar la sessió a l'usuari.
3. Consultar el llistat de factors: permet consultar ràpidament els factors de l'habitació on s'allotja l'usuari.
4. Consultar l'estat general dels factors: permet consultar de manera visual i gràfica en quin estat es troba cada factor i en quin percentatge d'optimització energètica es troba cada un dels factors que conformen la situació de l'habitació on s'allotja l'usuari.
5. Consultar les dades específiques de cada factor: permet consultar detalladament la informació específica del factor seleccionat.
6. Actualitzar informació dels factors: permet actualitzar la situació actual del sistema i per tant refrescar les dades dels diferents factors de l'habitació on s'allotja l'usuari amb les dades que NECADA recull.
7. Consultar estat general d'estalvi del sistema: permet consultar de manera ràpida el percentatge d'optimització total en què es troba l'habitació on s'allotja l'usuari.
8. Consultar l'estat de la situació: permet consultar la informació detallada que conforma la situació actual que es troba l'habitació on s'allotja l'usuari.
9. Consultar el perfil de l'usuari: permet consultar la informació sobre l'hotel i l'habitació on es troba allotjat l'usuari que té la sessió iniciada.
10. Consultar informació sobre l'aplicació: permet consultar la informació de creació de l'aplicació.

4.3. Diagrama de casos d'ús

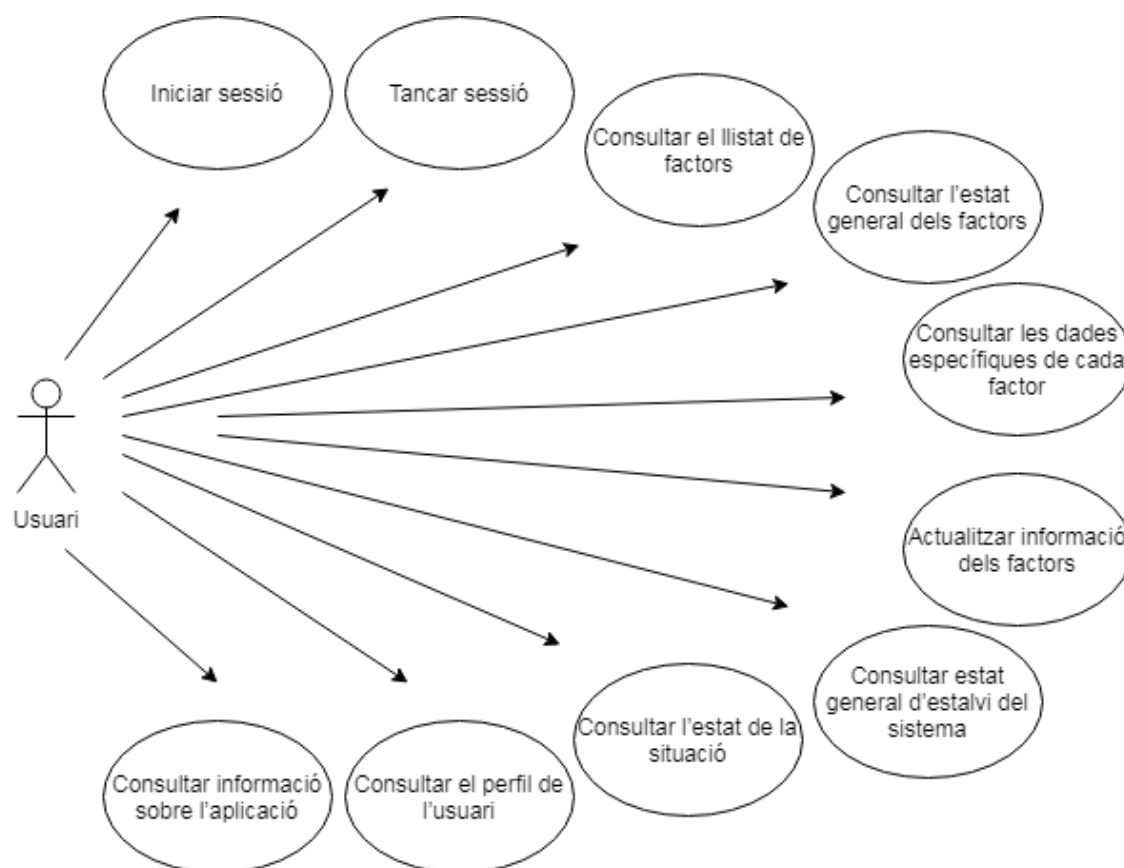


Figura 4: Diagrama de casos d'ús de l'aplicació

4.4. Casos d'ús

Ja definits aquests requisits funcionals que han de complir el projecte i mostrats en un diagrama, és hora d'explicar i especificar aquests casos d'ús de l'aplicació. Com que les funcionalitats que permet l'aplicació són poques, s'agruparan totes en un mateix apartat i posteriorment se'n desglossarà tota la informació:

Cas d'ús	1 - Iniciar sessió	Actor	Client de l'hotel
Precondició	L'usuari té unes claus que l'hotel li ha lliurat i un arxiu de validació d'aquestes. Aquest pot entrar-hi abans de la data de sortida de l'estància.		
Disparador	L'usuari vol accedir a l'aplicació.		
Escenari principal d'èxit			

<ol style="list-style-type: none"> 1. El sistema mostra la pàgina d'inici de sessió en cas de no haver-la iniciat prèviament. Si l'usuari ja té la sessió iniciada es passa al punt 4. 2. L'usuari introdueix la clau de l'habitació i la contrasenya. 3. L'usuari pressiona el botó de "Iniciar sessió". 4. El sistema comprova les dades amb l'arxiu facilitat per l'hotel i porta l'usuari a la pàgina principal.
Extensions
<ol style="list-style-type: none"> 4.1. Dades incorrectes. <ol style="list-style-type: none"> 4.1.1 El sistema adverteix a l'usuari que alguna de les dades és incorrecte. 4.1.2 El sistema torna al punt 1.

Taula 1: Cas d'ús "Iniciar sessió".

Cas d'ús	2 - Tancar sessió	Actor	Client de l'hotel
Precondició	L'usuari ja ha iniciat sessió prèviament.		
Disparador	El usuari ja no vol seguir amb la sessió iniciada.		
Escenari principal d'èxit			
<div>1. L'usuari prem la icona de tancar sessió.</div> <div>2. El sistema pregunta si realment n'estàs segur de tancar sessió.</div> <div>3. L'usuari confirma la tancada de sessió.</div> <div>4. El sistema redirigeix l'usuari a la pantalla de LogIn i en cas de voler tornar a veure les dades de l'habitació caldrà iniciar la sessió.</div>			
Extensions			
<div>3.1. L'usuari cancel·la la tancada de sessió.</div> <div>3.1.1 El sistema no redirigeix a cap pantalla i es manté a la mateixa on estava.</div>			

Taula 2: Cas d'ús "Tancar sessió".

Cas d'ús	3 - Consultar el llistat de factors	Actor	Client de l'hotel
Precondició	L'usuari està amb la sessió iniciada.		

Disparador	L'usuari vol veure els diferents factors.
Escenari principal d'èxit	
1. El sistema mostra una pantalla amb els diferents factors del sistema.	

Taula 3: Cas d'ús "Consultar el llistat de factors".

Cas d'ús	4 - Consultar l'estat general dels factors	Actor	Client de l'hotel
Precondició	L'usuari està amb la sessió iniciada.		
Disparador	L'usuari vol veure l'estat general dels diferents factors.		
Escenari principal d'èxit			
1. El sistema mostra una pantalla amb els diferents factors del sistema amb la informació general.			

Taula 4: Cas d'ús "Consultar l'estat general dels factors".

Cas d'ús	5 - Consultar les dades específiques de cada factor	Actor	Client de l'hotel
Precondició	L'usuari està amb la sessió iniciada i el sistema mostra una pantalla amb els diferents factors del sistema.		
Disparador	L'usuari vol veure el detall d'un dels factors que n'està veient la informació general.		
Escenari principal d'èxit			
1. L'usuari fa clic al factor que vol veure'n el detall. 2. El sistema mostra una pantalla amb tota la informació del factor.			

Taula 5: Cas d'ús "Consultar les dades específiques de cada factor".

Cas d'ús	6 - Actualitzar informació dels factors	Actor	Client de l'hotel
Precondició	L'usuari ha iniciat sessió i el sistema mostra una pantalla amb els diferents factors del sistema.		
Disparador	El client vol actualitzar la informació de l'habitació segons el moment actual.		
Escenari principal d'èxit			
1. L'usuari arrossega de dalt cap baix la pantalla principal. 2. El sistema mostra la informació actualitzada.			
Extensions			
2.1. No hi ha cap canvi respecte les dades anteriors i no es modifica res.			

Taula 6: Cas d'ús "Actualitzar informació dels factors".

Cas d'ús	7 - Consultar estat general d'estalvi del sistema	Actor	Client de l'hotel
Precondició	L'usuari està amb la sessió iniciada.		
Disparador	L'usuari vol veure l'estat general d'estalvi del sistema.		
Escenari principal d'èxit			
1. El sistema mostra una pantalla amb el percentatge total d'estalvi del sistema.			

Taula 7: Cas d'ús "Consultar estat general d'estalvi del sistema".

Cas d'ús	8 - Consultar estat actual de la situació	Actor	Client de l'hotel
Precondició	L'usuari està amb la sessió iniciada.		
Disparador	L'usuari vol veure la informació detallada que conforme la situació actual on es troba l'habitació.		
Escenari principal d'èxit			
1. L'usuari fa clic al botó de “Informació detallada de la situació” 2. El sistema mostra una pantalla amb la informació que fa referència a la situació actual.			

Taula 8: Cas d'ús "Consultar estat actual de la situació".

Cas d'ús	9 - Consultar el perfil de l'usuari	Actor	Client de l'hotel
Precondició	L'usuari està amb la sessió iniciada.		
Disparador	L'usuari vol veure informació de l'hotel i de l'habitació on s'allotja.		
Escenari principal d'èxit			
Android		iOS	
1. L'usuari fa clic als tres punts que apareixen a la part superior a la dreta. 2. L'usuari fa clic al botó "Perfil". 3. El sistema mostra una pantalla amb la informació de l'hotel i l'habitació on s'allotja.		1. L'usuari fa clic al botó "Perfil". 2. El sistema mostra una pantalla amb la informació de l'hotel i l'habitació on s'allotja.	

Taula 9: Cas d'ús "Consultar el perfil de l'usuari".

Cas d'ús	10 - Consultar informació sobre l'aplicació	Actor	Client de l'hotel
Precondició	L'usuari està amb la sessió iniciada.		
Disparador	L'usuari vol veure les dades relacionades amb el creador de l'aplicació.		
Escenari principal d'èxit			
Android		iOS	
<ol style="list-style-type: none">1. L'usuari fa clic als tres punts que apareixen a la part superior a la dreta.2. L'usuari fa clic al botó “Sobre l'aplicació”.3. El sistema mostra una pantalla amb les dades relacionades amb el creador de l'aplicació.		<ol style="list-style-type: none">1. L'usuari fa clic al botó “Sobre l'aplicació”.2. El sistema mostra una pantalla amb les dades relacionades amb el creador de l'aplicació.	

Taula 10: Cas d'ús "Consultar informació sobre l'aplicació".

4.5. Requisits de qualitat

Per altra banda, qualsevol sistema també integra una sèrie de requisits que es refereixen a que les característiques del software o el funcionament d'aquest tinguin unes certes propietats de qualitat. A continuació es llisten els diferents requisits de qualitat segons Volere [47] tal i com es va aprendre a fer a l'assignatura d'Enginyeria dels Requisites:

Requisit #	1	Tipus de requisit segons Volere:	10.a Appearance
Descripció	El disseny de l'aplicació serà atractiu per tota classe d'usuaris.		
Justificació	Es vol que els usuaris contribueixin a l'estalvi energètic de l'hotel i per tant l'experiència amb l'aplicació ha de ser agradable.		
Condicció de satisfacció	Que un 80% dels usuaris que l'hagin utilitzat ho hagin fet d'un principi a un final i no l'hagin deixat d'utilitzar.		

Taula 11: Requisit Volere "10.a Appearance".

Requisit #	2	Tipus de requisit segons Volere:	11.a Ease of use
Descripció	El disseny de l'aplicació serà d'ús senzill per tota classe d'usuaris.		
Justificació	Es vol que qualsevol persona es senti còmode utilitzant aquest sistema i no deixi d'utilitzar-la.		
Condicció de satisfacció	Que un 80% dels usuaris que l'hagin utilitzant valorin positivament el sistema.		

Taula 12: Requisit Volere "11.a Ease of use".

Requisit #	3	Tipus de requisit segons Volere:	11.c Learning
Descripció	Les accions que es poden dur a terme al sistema es podran aprendre fàcilment.		
Justificació	Com que es vol que l'usuari utilitzi l'aplicació serà necessari que aquest pugui aprendre ràpidament com utilitzar-la.		
Condicció de satisfacció	En cas de que no es rebin queixes en els hotels on s'utilitzin o els usuaris utilitzin correctament l'aplicació podrem dir que es satisfà.		

Taula 13: Requisit Volere "11.c Learning".

Requisit #	4	Tipus de requisit segons Volere:	12.d Reliability and Availability
Descripció	El nivell de disponibilitat del servidor serà superior al 99%.		
Justificació	Es necessita que el servidor estigui disponible al màxim de temps possible per evitar interrupcions de servei a els usuaris.		
Condicció de satisfacció	El servidor on està allotjat el sistema NECADA, que és d'on s'extrauran les dades, en l'aplicació definitiva no és un factor crític la disponibilitat i en aquest aspecte està relaxat i es pot assegurar una disponibilitat total.		

Taula 14: Requisit Volere “12.d Reliability and Availability”.

Requisit #	5	Tipus de requisit segons Volere:	12.e Robustness or FaultTolerance
Descripció	L'aplicació continuarà funcionant en el dispositiu si es perd la connexió a Internet (simplement no es podran actualitzar les noves dades de percentatge dels diferents factors).		
Justificació	Tot i que l'usuari no pugui veure la millora dels canvis que pot produir a la situació, necessita tenir accessible la informació que havia obtingut per últim cop.		
Condicció de satisfacció	<p>Es corroborarà que l'aplicació segueixi funcionant sense connexió:</p> <ul style="list-style-type: none"> • El usuari podrà veure la última informació general i específica del sistema que s'ha obtingut prèviament sempre que no hagi tancat la sessió durant la desconexió. • Fins que no hi torni haver connexió a Internet, l'usuari no podrà obtenir dades actualitzades de nou. 		

Taula 15: Requisit Volere “12.e Robustness or FaultTolerance”.

Requisit #	6	Tipus de requisit segons Volere:	12.g Scalability or Extensibility
Descripció	El sistema estarà preparat per incorporar altres funcionalitats similars o afegir característiques.		
Justificació	Ha de ser escalable ja que l'aplicació s'anirà actualitzant a mesura que apareguin noves idees i funcionalitats.		
Condicció de satisfacció	El sistema s'ha implementat orientat a l'escalabilitat podent canviar les característiques d'algunes funcionalitats o per afegir funcionalitats sense afectar a les funcionalitats ja implementades.		

Taula 16 : Requisit Volere “12.g Scalability or Extensibility”.

Requisit #	7	Tipus de requisit segons Volere:	14.c Adaptability
Descripció	L'aplicació s'executarà a Android i iOS de manera nativa.		
Justificació	Ja que aquestes plataformes són les més comuns actualment, se'ls hi donarà suport a totes elles.		
Condicció de satisfacció	L'aplicació funcionarà tan en Android com en iOS.		

Taula 17: Requisit Volere "14.c Adaptability".

Requisit #	8	Tipus de requisit segons Volere:	15.c Privacy
Descripció	El tractament de les dades de l'habitació que siguin registrats en el sistema es faran d'acord amb la legislació vigent.		
Justificació	La protecció de dades personals és un fet fonamental regulat per la legislació i que, per tant, s'ha de respectar.		
Condicció de satisfacció	El tractament de dades de caràcter personal que fa el sistema respecte íntegrament la legislació vigent. En aquests moments, consisteix en el Real decret 1720/2007 [48], de 21 de desembre, pel qual s'aprova el reglament de desenvolupament de la Llei orgànica 15/1999, de 13 de desembre, de protecció de dades de caràcter personal.		

Taula 18: Requisit Volere "15.c Privacy".

4.6. Esquema conceptual

En aquest apartat es dedica a presentar les diferents classes que formen els models de l'aplicació i a mostrar-les gràficament amb un diagrama de classes.

4.6.1. Diagrama de classes UML

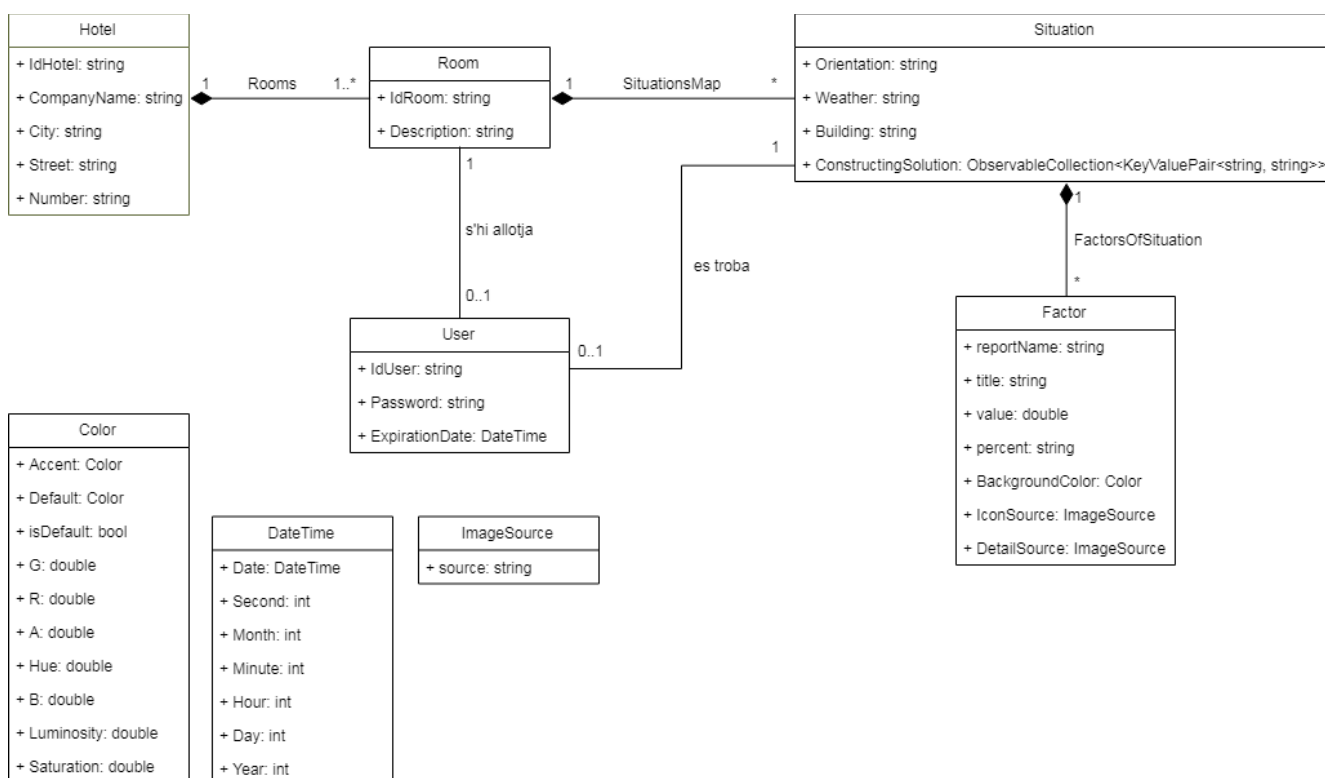


Figura 5: Diagrama de classes UML de l'aplicació.

4.6.2. Explicació de les classes, atributs i associacions

- **Hotel:** un hotel és una entitat que representa un hotel on es pot allotjar un dels possibles clients. Un hotel té una *clau que l'identifica* (IdHotel), un *nom* (CompanyName), una *ciutat* (City), un *carrer* (Street) i un *número de carrer* (Number). També conté una o moltes *habitacions* (Rooms).
- **Room:** una habitació és una entitat que representa una habitació d'un hotel en concret. Una habitació té una *clau que l'identifica* (IdRoom), una *descripció* (Description) i una *referència interna a l'hotel* del qual forma part (FromHotel). També conté cap o moltes *situacions* diferents on es pot trobar (SituationsMap).
- **Situation:** una situació és una entitat que representa una situació concreta que defineix l'estat en què es troba una habitació. Una situació té una *orientació* (Orientation), un *temps* (Weather), un *tipus d'edifici* (Building) i varies *solucions de construcció* (ConstructingSolution). També conté cap o molts *factors* que conformen la situació de l'habitació.
- **Factor:** un factor és una entitat que defineix una recol·lecció de dades concreta de la situació en qüestió com podria ser el consum d'energia elèctrica interior de l'habitació. Un factor té un *nom que l'identifica* (reportName), un *títol* (title), un

valor (value), un *percentatge* (percent), un *color de fons* (BackgroundColor), una *icona* identificativa (IconSource) i una *imatge de detall* concreta (DetailSource).

- User: un usuari és l'entitat que representa un client que s'ha allotjat en una habitació d'un hotel que disposa del sistema i se li ha facilitat la clau d'entrada a l'aplicació amb la contrasenya corresponent. Un usuari té una *clau d'identificació* que està formada per la clau d'identificació de l'hotel i habitació on s'allotja (IdUser), una *contrasenya associada al compte* (Password), i la *data de sortida de l'hotel* (ExpirationDate). L'usuari és l'element que dona accés a l'aplicació.

5. Desenvolupament del projecte

5.1. Arquitectura del sistema

L'arquitectura de l'aplicació dissenyada s'explica anteriorment, definint en quines capes es divideix el codi, però no s'ha mostrat tot el codi resultant repartit en les diferents carpetes. Aquí tenim els diferents models, vistes i les vistes model del el projecte:

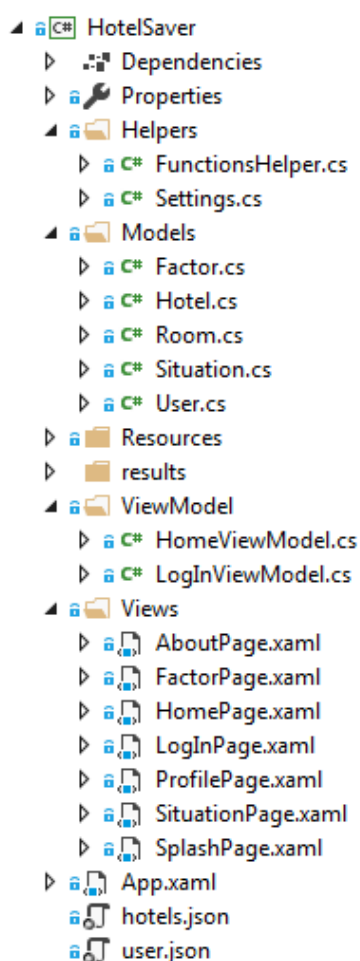


Figura 6: Fitxers principals que componen l'aplicació.

La carpeta de “Resources” conté els Embedded Resources que s'utilitzen per l'aplicació, la carpeta “results” és la carpeta que conté totes les dades de prova de NECADA, i “Helpers” conté dos fitxers que contenen diferents funcionalitats que l'aplicació necessita per poder dur a terme totes les seves funcionalitats. La resta de carpetes ja han estat explicades amb l'explicació del patró MVVM.

També s'ha de tenir en compte que també formen part del projecte les carpetes que tenen el codi concret de Android i iOS.

L'arquitectura esperada del sistema de l'aplicació resultant té una estructura una mica més complexa que la de l'aplicació amb la seva estructura interna, també s'hauria de comunicar amb l'exterior. Aquí es mostra un gràfic que mostra quines peticions hauria de poder dur a terme l'aplicació i com simula ara per ara aquestes connexions:



Figura 7: Arquitectura del sistema i de la solució prototipada.

Aquest esquema que mostra on es connecta l'aplicació està més detallat a l'apartat que es detalla el disseny de la BBDD a l'apartat 5.2.3.

5.2. Disseny

Primer de tot cal comentar que el disseny pel que fa la interfície i el software de l'aplicació té una particularitat especial, i és que al fer l'aplicació multiplataforma amb Xamarin

(nativa), et permet tenir un codi base .NET compartit per les diferents plataformes (Android i iOS), i després per separat el codi explícit per cada sistema operatiu. L'objectiu sempre va ser que en cas de que és pogués utilitzar el codi compartit, fer-ho, per no haver d'implementar una mateixa funcionalitat en cada sistema operatiu. [49]

5.2.1. Disseny de la interfície

5.2.1.1. Disseny de l'estructura

En aquest treball el disseny de la interfície de l'aplicació va ser un dels elements bàsics. Per començar es tenia molt clar quin tipus de dades es podrien mostrar a l'usuari, però calia pensar correctament quina era la millor manera de mostrar-les-hi. Així que imaginar com seria la pantalla de la funcionalitat principal va ser una de les tasques fonamentals per començar a traçar el que seria l'aplicació.

5.2.1.1.1. Vista principal

Primer de tot es va fer un mock-up a mà, esbossant com es podrien mostrar les dades de forma simple i en cas de voler-ne obtenir més detall quina podia ser la manera. En aquell moment es sabia que es tindrien uns factors diferents que podien estar correctament alineats amb la situació ideal, que aquests podrien contenir més informació, i que haurien d'ajudar a aconseguir una situació ideal en cas de no ser-hi.

La idea en la que es volia treballar era una pantalla que oferís a l'usuari la ràpida visualització de les dades, i en cas de que aquesta situació no sigues la més adient poder veure quins factors es podrien modificar perquè aquesta situació esdevingués la ideal. La idea principal que es volia aplicar sobretot va ser la de mostrar les dades de manera intuïtiva i clara, ajudant-se amb colors per reforçar la indicació de si el sistema està correctament optimitzat o podria millorar. El sistema que s'utilitzaria per mostrar com de correctament estava cada factor era amb colors de semàfor.

Per poder estructurar totes aquestes vistes es va utilitzar sempre el XAML sense utilitzar per a aquest fi el **code-behind** que té associat cada vista. Per poder aplicar el tipus de llistat que es volia utilitzar es va veure que no existia cap tipus de vista que permetés llistar diferents instàncies d'un model i col·locar-les en files i columnes. Aquesta vista havia de permetre aquesta entrada d'elements com permetia la ListView però col·locar-los com podria permetre un Grid, però no hi havia cap opció integrada al sistema viable. Després de temps buscant es va trobar com s'ha mencionat anteriorment la FlowListView, que era una vista que derivava de la ListView i a més a més permetia posar els elements per columnes tal i com es volia. Un cop amb aquests diferents elements en joc ja es podia estructurar la pantalla principal tal i com s'havia imaginat.

Finalment així és com va quedar la interfície de la vista principal de l'aplicació:



Figura 8: Vista principal de la versió Android



Figura 9: Vista principal de la versió iOS

Aquesta Vista pot mostrar altres elements que en un estat normal no els mostrarà: És per exemple el cas de diferents Alert Dialogs que apareixen si l'usuari fa clic al botó d'enrere (preguntant-li si realment vol sortir-ne), quan l'usuari aconsegueix per primer cop superar el 90% d'optimització general, i quan aquest vol tancar sessió (fent clic al botó que hi ha a la part superior dreta de la pantalla) que li pregunta altra vegada si realment vol tancar-la.

5.2.1.1.2. Vista de LogIn

Per fer el disseny de la vista de LogIn es va tenir en compte simplement que es volien tenir dos camps per introduir dades, el id de l'usuari i una contrasenya. Senzillament es va afegir el logotip de l'aplicació a la part superior perquè li donés continuïtat a l'usuari que per primer cop utilitza l'app.

La Vista també integra sota el botó d'inici de sessió un contenidor que en cas d'haver-hi error en les dades que s'introdueixen al formulari, s'ompliria amb el missatge d'error que s'hagués capturat

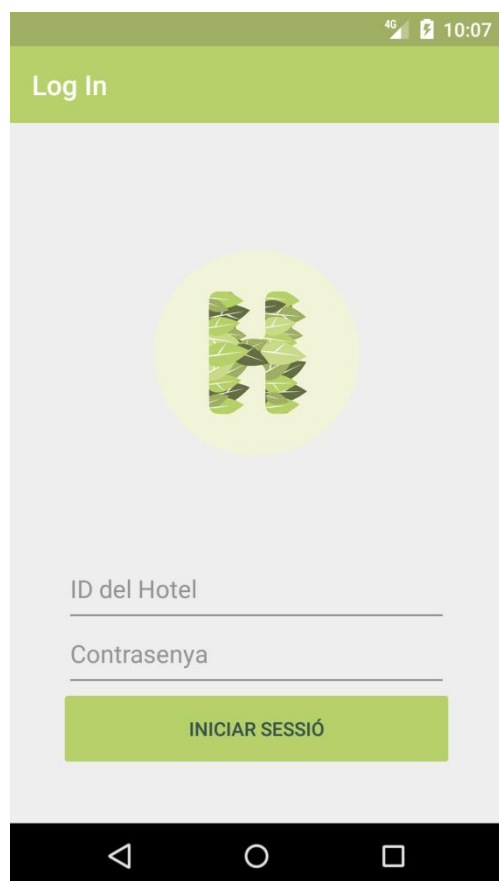


Figura 10: Vista d'inici de sessió de la versió Android



Figura 11: Vista d'inici de sessió de la versió iOS

5.2.1.1.3. Vista detall de factor

Per ajudar a que l'aplicació tingués un ús intuïtiu, en fer clic a un factor del llistat de la pàgina principal, n'apareix la informació d'aquell factor, ajudant a l'usuari a saber què cal fer per optimitzar la situació en que es troba si aquesta ho requerís.

En aquesta Vista es va creure necessari seguir insistint en el color que defineix com de bé està el percentatge, així l'usuari pot saber simplement amb un cop d'ull si està optimitzant correctament el factor seleccionat. També hi apareix una breu explicació de què tracta el factor, quines dades recull i engloba. Amb la informació que proporcionava la versió de proves de NECADA no es podia informar més sobre un factor en concret, així que aquesta va ser la informació que va acabar apareixent en aquesta vista i és així com va quedar:



Figura 12: Vista del detall del factor de la versió Android



Figura 13: Vista del detall del factor de la versió iOS

5.2.1.1.4. Vista detall de la situació

Aquesta vista mostra les dades que fan referència a les variables que componen la situació actual. Aquesta pàgina havia de ser simplement informativa, mostrar les dades d'alguna manera perquè es poguessin veure. D'un bon principi es va pensar introduir-les a la pantalla principal però es va pensar que estaria massa carregada i era millor introduir-les en una pàgina plana sense cap mena de disseny a part sinó que fos simplement informativa. Per tant finalment a la pàgina principal simplement hi va quedar un botó per poder accedir a la informació explícita de la situació actual en que es troba l'aplicació.

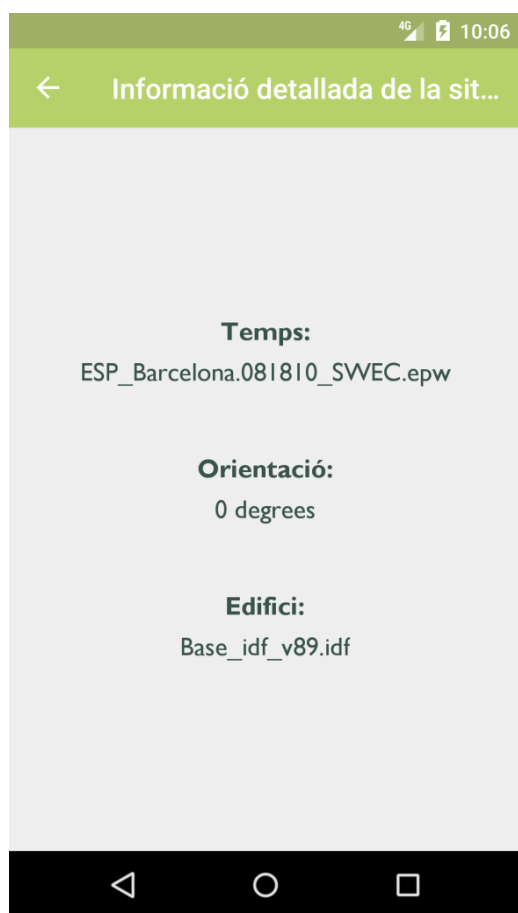


Figura 14: Vista del detall de la situació de la versió Android

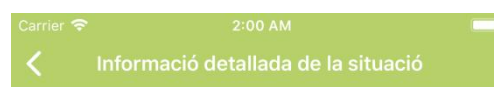


Figura 15: Vista del detall de la situació de la versió iOS

5.2.1.1.5. Vista de perfil

A la vista de perfil s'hi pot accedir a través del menú desplegable que s'obra fent clic als tres punts que apareixen a la part superior de la pantalla d'inici. En aquesta s'hi mostra a l'usuari informació sobre l'hotel on està allotjat i una descripció sobre l'habitació que li correspon. A més a més, també mostra la data en que s'ha de deixar l'habitació: en cas que la data de sortida sigui l'endemà de la data actual, aquest container apareixerà de color vermell per indicar més visualment que cal tenir-ho en compte.

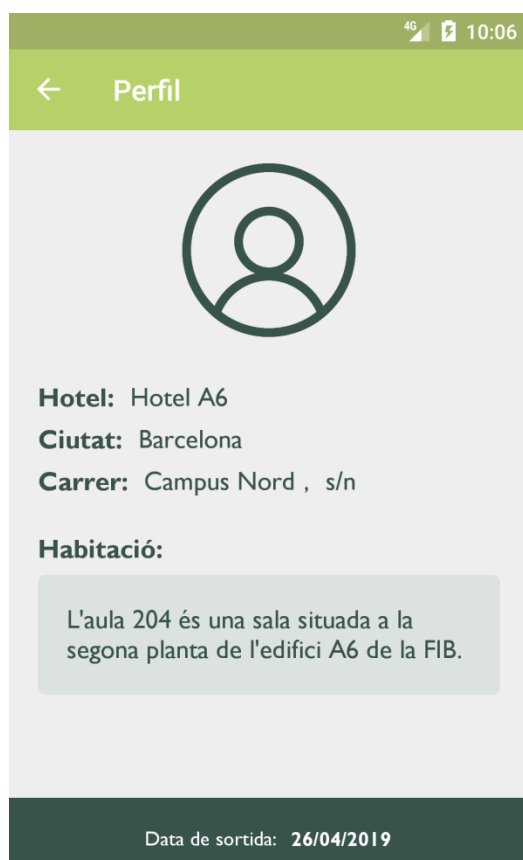


Figura 16: Vista del perfil de la versió Android

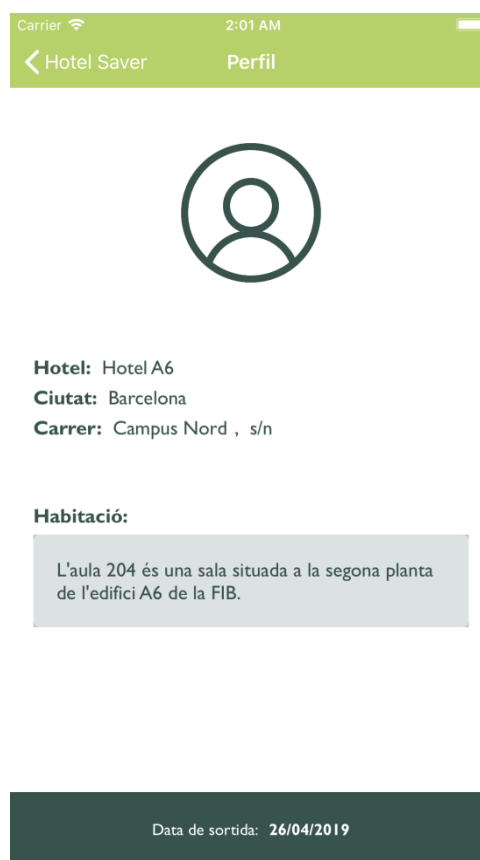


Figura 17: Vista del perfil del factor de la versió iOS

5.2.1.1.6. Vista de informació de l'aplicació

Aquesta és una vista simplement informativa que serveix per mostrar les dades que fan referència a l'autoria de l'aplicació. S'hi accedeix a través del menú desplegable que també s'utilitza per entrar al perfil com hem vist en el punt anterior.



Figura 18: Vista de la informació de l'aplicació de la versió Android

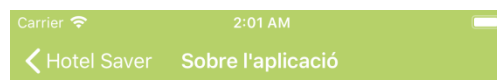


Figura 19: Vista de la informació de l'aplicació de la versió iOS

5.2.1.1.7. Vista splash

Aquesta vista és la primera de totes que apareix quan s'inicia l'aplicació. La idea d'aquesta vista és preparar-nos per l'aplicació mentre s'estan executant algunes funcions necessàries pel funcionament de l'aplicació. A més a més de mostrar el logotip, per insistir amb la marca del projecte, també s'hi mostra la informació més bàsica de la creació de l'aplicació. El logotip fa un efecte de batec que vol reproduir aquesta vivacitat de l'aplicació.



Figura 20: Vista del splash de la versió Android

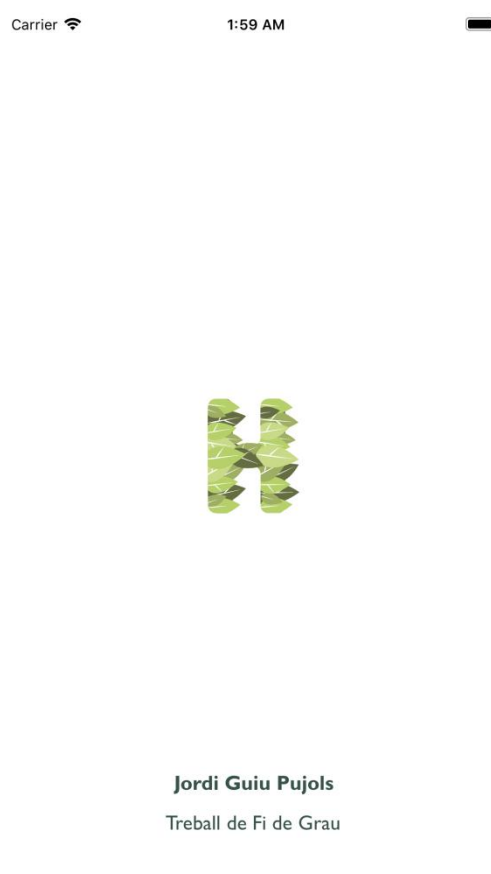


Figura 21: Vista del splash de la versió iOS

5.2.1.2. Colors

Per a la creació d'aquest projecte de bon principi es tenia la idea al cap que els colors que havien de compondre-la havien de ser tons verds abocant a l'ecologisme i a la natura, que és un dels aspectes principals pels quals es lluita des d'aquesta aplicació.

Primer de tot, per tenir un disseny lo més simple possible i simplificar el missatge que es volia donar es va creure que era convenient utilitzar **Flat Design** [50].

En aquests tipus de dissenys es solen utilitzar colors pastel amb una cromàtica bastant similar entre colors sense fer uns salts molt exagerats. També és molt habitual utilitzar aquests

colors per a ajudar a l'usuari a entendre certa informació: com ja s'havia pensat amb els colors del semàfor de les caixetes per mostrar a l'usuari com de bé estava optimitzat un cert factor.

Així doncs aquest tipus de disseny anava perfecte per l'aplicació i connectava totes les coses amb les quals es volia treballar: simplicitat de l'aplicació, de fàcil aprenentatge i amb una aparença molt agradable.

Es van decidir una paleta de tons verds pastel per començar a treballar amb la paleta que majoritàriament formaria part de l'aplicació:



Figura 22: Paleta dels colors verds utilitzats a l'aplicació

Aquests colors es van utilitzar de bon principi per la part superior del menú.

L'aplicació requeria tres colors, de semàfor, per indicar com de bé estava cada factor en la situació del sistema. Així doncs es van generar els altres dos colors per indicar que la cosa no estava del tot bé i que estava malament, també amb tons pastel:



Figura 23: Paleta dels colors taronges utilitzats a l'aplicació



Figura 24: Paleta dels colors vermells utilitzats a l'aplicació

Amb aquests colors ja podíem il·lustrar gràficament en quin estat estava un factor en concret embolcallant-lo d'un dels colors (decidint un to més clar). El color taronja també es va afegir com a **colorAccent** de l'aplicació ja que aquest es mostra en certes accions d'alertes i aquest és un color més neutre que els altres dos i a més a més les persones el tenim associat a les indicacions d'alerta i fa estar més atents.

Després de tot això es va veure que el color negre, que s'utilitzava pel text, quedava massa dur i sobresaltava massa dels diferents colors pastel que definien la paleta de l'aplicació. Per tant es va buscar un to blau grisós que s'utilitzés per a aquest fi.



Figura 25: Paleta dels colors blau grisosos utilitzats a l'aplicació

Així doncs, quedava una paleta on el verd i el vermell eren complementaris entre ells i un taronja i blau grisós que també eren complementaris entre ells: **harmonia complementària**. A més els colors de cada fila entre ells quedaven amb una **harmonia d'anàlegs**, per tant es va creure que els colors que quedaven a la paleta eren ideals per combinar-los entre si i que el resultat quedés bonic.

5.2.1.3. Icones

Primer de tot cal mencionar que per dissenyar aquestes icones es va comptar amb l'ajuda de la Mar Guinart que va donar un cop de mà desinteressadament dibuixant les diferents figures que eren necessàries per il·lustrar cada factor.

Aquestes icones es van dissenyar també amb **flat design** i amb els colors que conformaven la paleta, que s'ha explicat en l'apartat anterior, tenint en compte alguna limitació que es va veure més endavant com s'explica a continuació. Si els icones contenien algun dels colors de la paleta de colors vermells donava massa negativitat al global de la caixa, i tot i que estigués verda i el percentatge es veiés correcte donava la sensació que no estava del tot bé per culpa d'aquest color de la icona. Així doncs, es va decidir utilitzar els altres colors per dissenyar els icones ja que quedaven tots més neutres i no destacaven tant com ho feia el vermell. Potenciant l'ús del taronja i verd també tenia sentit tenint en compte que un era el color principal de l'aplicació i l'altre el **colorAccent** i això els hi donava un sentit més global i estret amb l'aplicació en si mateixa.

Aquí es presenten les dotze icones que es van dissenyar pels dotze factors que existien en la versió de proves proporcionada per NECADA:



Figura 26: Icones utilitzades pels dotze factors a l'aplicació

No és un aspecte gaire rellevant, però també cal mencionar que el nom de l'aplicació que es va escollir va ser “Hotel Saver” ja que era un nom que no estava agafat ni a la botiga d'aplicacions de Google ni a la d'Apple, i declarava bé les intencions d'aquesta aplicació.

Finalment s'havia de dissenyar el logotip de l'aplicació i aquest es va crear al final de tot amb la paleta dels verds de l'aplicació també simbolitzant l'ecologisme i la natura amb aquests tons i les fulles que conformen la lletra “H” d'hotel. Per utilitzar les mides recomanades per a Android es va utilitzar una eina en línia [51] en canvi per iOS es va poder introduir directament des de Visual Studio per les mides exactes que demanava.

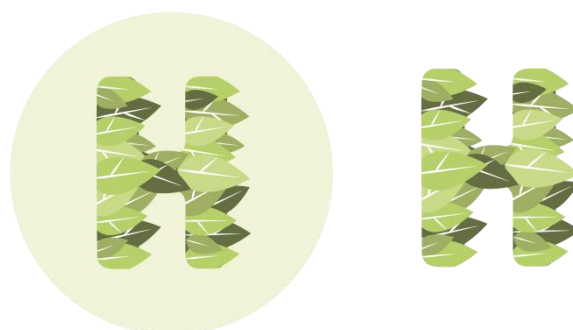


Figura 27: Logotip de l'aplicació amb fons i sense

5.2.1.4. Tipografia

Quan es va dissenyar l'estructura de la pàgina principal amb els icones adients es va pensar en que calia posar una tipografia especial i no simplement la que venia per defecte amb l'aplicació, ja que aquesta també podia jugar un paper important en com es veia l'aplicació.

La font que es va utilitzar no va respondre estrictament a uns criteris parametritzables sinó que es va regir a criteris estètics purament objectius. Es va escollir la famosa Gill Sans MT, i se n'utilitzava les variants depenent de quin era el cas en què s'utilitzava. Si es volia destacar un text que per posicionament no quedava tan ressaltat, s'utilitzava la versió en negreta. Si es volia mostrar un text com a títol d'un apartat, es mostrava el text en quèstió

en majúscula i a més a més s'utilitzava la lletra estreta que ajudava a donar-li èmfasi al dotar-la de sofisticació i a més a més s'estalviava espai possible que podria ocupar el títol en una caixa que no és gaire gran (per la mida dels llocs on s'utilitzava no abocava a la confusió, com a vegades passa amb aquest tipus de lletra) [52]. En última instància si en un text normal s'utilitzava la versió bàsica de la tipografia, això feia que la resta d'elements pel simple fet de tenir una lletra diferent destacaven per sobre de la majoria d'elements.

5.2.1.1. Diagrama de navegabilitat

En aquest apartat es vol mostrar el diagrama de pantalles de l'aplicació, de quina vista es pot anar a l'altra. La navegabilitat és una part molt important d'aquesta, ja que una bona experiència d'usuari pot ser clau perquè l'usuari utilitzi aquesta aplicació. Com es va pensar de bon principi l'aplicació, havia de ser el màxim d'intuitiva perquè la poguessin utilitzar tot tipus d'usuaris i com es pot veure les funcionalitats bàsiques s'hi poden accedir des de la pantalla d'inici, sent aquesta on es centra tota l'activitat:

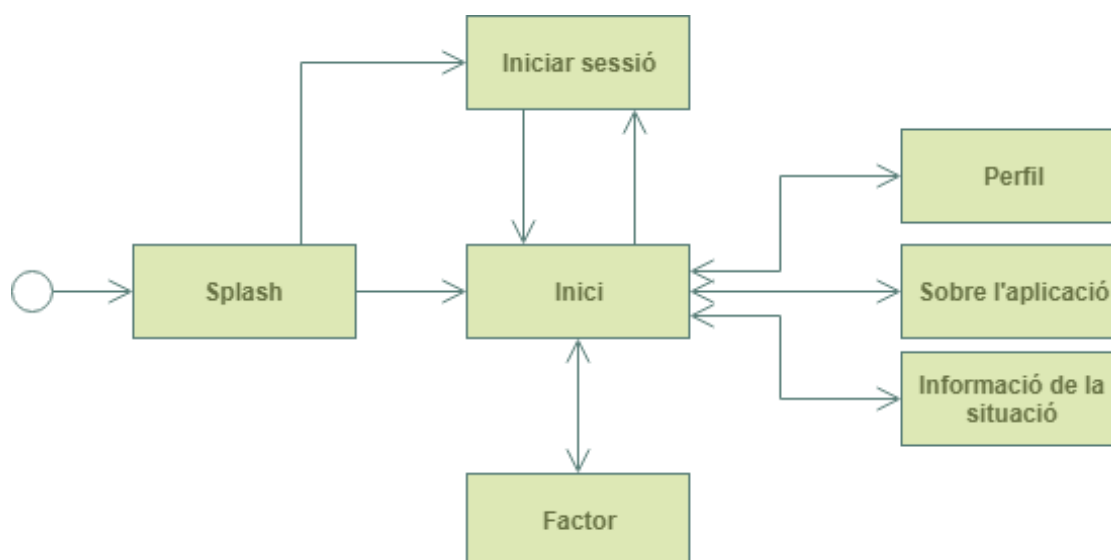


Figura 28: Diagrama de navegabilitat de l'aplicació

Prèviament ja s'han definit els diferents casos d'ús i per tant és allà on està definit quina acció s'ha de prendre per anar d'una vista a una altra depenent de l'acció que vulgui dur a terme l'usuari.

5.2.2. Disseny del software

Al llarg de la carrera, i sobretot en l'especialitat d'Enginyeria del Software, s'ha fet molt èmfasi que a l'hora de programar és molt important utilitzar certs patrons perquè el codi sigui reutilitzable i fàcilment comprensible per a tercers.

El patró general que s'ha utilitzat en aquesta aplicació ha estat el MVVM (explicat més detalladament al punt 3.4. d'aquest treball).

En la capa de dades, als Models, hi ha tota la lògica de l'aplicació, com són les classes: *User*, *Factor*,... Per altra banda, les Vistes són l'estructura, el disseny, l'aparença que l'usuari veu a la pantalla del mòbil i amb la qual interactua. A les ViewModels es donen agrupen les funcionalitats que les vistes necessiten per poder dur a terme les diferents tasques.

Totes aquestes característiques que aporta el MVVM fan molt servei per desenvolupar una aplicació amb Xamarin.Forms ja que implica crear unes interfícies d'usuari afegint codi per darrere (**code-behind**) que opera en aquesta interfície. A mesura que l'aplicació es modifica i creix, el manteniment es podria convertir en un greu problema, i com més lligada estigues la interfície de la lògica suposaria més canvis a cada capa, i a més a més, greus dificultats en poder provar el codi. Amb aquesta solució es podrà treballar en cada aspecte més independentment i sense problemes de sincronització entre fitxers i canvis.

La gran majoria de codi d'aquesta aplicació va ser desenvolupat a la part de codi compartit, així no s'havia de repetir codi que feia la mateixa funcionalitat per les dues plataformes. Durant el treball doncs, un dels principals objectius era implementar tantes funcionalitats com fos possible al codi compartit minimitzant tant com fos possible haver d'utilitzar codi concret per a cada plataforma.

La major part de cops es va aconseguir mitjançant llibreries com ja s'ha explicat anteriorment, però també hi va haver altres aspectes que no hi va haver més remei que haver-ho d'implementar per cada sistema operatiu per separat. Va ser el cas dels següents elements:

- Icona de l'aplicació: Tant iOS com Android obtenen el logotip de arxius locals i per poder-lo definir va caldre incloure'ls dins de cada carpeta concreta dels diferents sistemes operatius.
- Color de la barra superior: Els colors de les barres superiors en el cas de Android, i el color general de la barra superior de iOS també es van introduir de diferents maneres. Android utilitza el `colorPrimary` i el `colorPrimaryDark` que es defineixen en arxius interns per a definir els colors de l'aplicació. Pel cas de iOS aquests colors es poden definir de diferents maneres, però la que es va utilitzar va ser inserint unes propietats concretes en el moment de la creació de les pàgines de navegació que permetia escollir el color del fons de la barra superior i el color del text.
- Tipografia: Les tipografies que s'han utilitzat es van definir conjuntament en el codi compartit però es van haver d'inserir explícitament a dins de cada codi dels sistemes operatius.

Són molts pocs els casos que es va haver de modificar el codi concret de cada sistema operatiu i això diu molt de la potència real que té Xamarin podent utilitzar casi totalment el codi compartit i genera una aplicació nativa pels diferents sistemes sense cap problema.

Un problema que té l'aplicació és que quan s'hi fa clic des del menú del dispositiu triga un temps fins que no apareix la pàgina de splash. Normalment estem acostumats que aquesta

transició entre seleccionar una aplicació i que aparegui la primera pàgina, sigui instantània. Un cop es fa clic en aquesta aplicació es veurà que hi ha una pàgina en blanc durant uns segons i després ja apareix la primera de totes, la pàgina de splash.

El cas és que el sistema ha de carregar moltes **assemblies** i això porta temps, i encara més en simuladors i executant la versió de **Debug**. Hi ha manera d'aconseguir que alguns d'aquests processos interns s'executin més tard i per tant no es vegi afectada l'aplicació aparentment, però són processos costosos per guanyar realment part d'aquests segons i per aquest prototip no era la cosa més important de totes.

El que s'ha de tenir clar és que la utilització de Xamarin implica que el pes de l'aplicació sigui gran i a més a més el temps de càrrega bastant dolent. S'ha de tenir en compte que permet acostar-se al màxim a l'arquitectura d'una aplicació nativa i això és quelcom molt costós. Xamarin ha de fer crides a través d'altres sistemes per executar funcions natives disponibles en els sistemes operatius que vol executar nativament, i això no és quelcom trivial i ràpid. Cada sistema de desenvolupament d'aplicacions multiplataforma que volen aconseguir aplicacions natives són incompletes i tenen els seus defectes. Per tant del què es tracta es de sospesar aquests inconvenients i avantatges i veure que convé tenint en compte totes les característiques [53].

Per altra banda també és important explicar les funcionalitats més importants que implementa l'aplicació. A continuació s'explicaran en més detall aquestes funcionalitats per saber més exactament com funciona el prototip desenvolupat:

- LogIn(): La funció d'iniciar sessió comprova primer de tot que no hi hagi ni l'identificador de l'usuari ni la contrasenya introduïdes des del formulari buides, i després ja fa les comprovacions de que aquestes dades introduïdes corresponguin a les dades de la clau virtual que l'hotel ha proporcionat a l'usuari. Si coincideixen, el sistema busca en quin hotel i habitació s'allotja el client ja que la clau que se li lliura és la composició de la clau de l'hotel i de l'habitació on està assignat. Més endavant a la secció del disseny de la BBDD s'explica com es faria perquè no hi hagués col·lisions d'identificadors. Un cop troba la clau a la base de dades l'usuari inicia la sessió i el seu identificador és emmagatzemat al Settings. També s'hi emmagatzema la data de sortida de l'estància del client, ja que podria passar que l'usuari no hagués tancat la sessió i seguis accedint a les dades de l'habitació, però per evitar-ho es comprova que la data de finalització sigui menor a la data actual. Si en algun cas d'aquests no es troba la informació que es busca a la base de dades d'hotels o a la clau virtual, el sistema mostra un missatge d'error.
- valuesToPercent(): Aquesta funció el que fa és transformar el valor que té associat el factor en un percentatge. Això ho fa recollint prèviament els valors màxims i mínims d'un mateix factor en totes les situacions possibles, per tant llavors es sap entre quins valors pot estar el factor en concret. Un cop amb aquestes dades genera un percentatge del 0% al 100% amb el valor que té comparant-lo amb els valors màxims i mínims. En cas de que un factor tingui els mateixos valors mínims i màxims en totes les situacions possibles, es va decidir amb el director de mostrar el factor amb un 100% significant que el factor

esmentat no aplica i no es possible modificar-ne els valors en totes les situacions. En el cas de proves hi ha molts factors que els hi passa això mateix ja que les dades no són molt canviants. Degut a això es va capgirar el llistat de factors simplement perquè els primers factors mai canviaven de valors i semblava que visualment no canviés l'aplicació, va ser un canvi senzillament per fer més fàcil la demostració del què es fa però el funcionament segueix sent el mateix però amb un ordre de llistat diferent.

- readSituation(): Quan s'executa per primer cop la pàgina d'inici de l'aplicació i sempre que es refresca s'executa aquesta funció, que el que fa és llegir de les dades de NECADA i posar-les en un llistat de les situacions actuals disponibles. Per decidir en quina situació està l'habitació en un precís moment no es pot decidir en aquest projecte, ja que NECADA no n'informa, per tant la decisió que es va prendre amb el professor va ser que el sistema escollís una de les situacions aleatòriament cada cop que s'actualitza l'aplicació, així es pot veure com es modifiquen les dades quan canvia la situació. El primer cop que s'executa es va fer que sempre agafés la primera situació de les que es llegeixen dels documents de prova de NECADA. Les situacions estan relacionades amb les habitacions de cada hotel, però en el cas de l'aplicació de prototip no tenia sentit utilitzar les situacions de la habitació on estava allotjat l'usuari, ja que només es tenien unes dades de situacions que són les que es van utilitzar per qualsevol cas. Cal dir que està preparat per funcionar en diversos hotels i habitacions però de cara un cas futur s'haurien de modificar lleugerament certes coses com els diccionaris que s'utilitzen per associar les claus dels factors amb les descripcions completes de cada un, es van inserir totes les possibilitats amb les dades de prova que es tenien.

És molt important doncs tenir en compte que l'aplicació funciona perfectament per les funcionalitats que es volien integrar i amb les dades que es van obtenir de NECADA, simplement que algunes funcionalitats que en cas de l'aplicació definitiva s'haurien d'implementar no tenia sentit fer-ho ja que no es podia aplicar de cap manera al resultat final del treball.

5.2.3. Disseny de la BBDD

Pel que fa a les bases de dades que es van utilitzar en el projecte tenen algunes particularitats especials que s'explicaran a continuació.

Al no connectar-se al sistema de base de dades de NECADA ja que això suposava unes tasques addicionals que no entraven a l'abast del projecte que es volia dur a terme, es va decidir conjuntament amb el director del projecte d'agafar unes dades de prova puntuals d'unes situacions concretes d'una simulació d'una habitació d'un edifici de prova. Va ser el mateix director que va proporcionar aquests fitxers.

Per tant les dades de les situacions i dels factors de cada situació es tenien en uns fitxers en local a dins de l'aplicació, que l'aplicació consulta per mostrar les dades pertinents. Aquest fitxer local té la mateixa estructura que tenen els fitxers generats per NECADA amb totes les dades de la situació i dels factors, per tant, extreure'n correctament les dades i modelar-les

permetia no només extreure les dades de prova amb que es volia treballar sinó tenir el model d'extracció de dades de cara un futur quan l'aplicació s'hagués de connectar a temps real amb el sistema de gestió d'energia.

NECADA extreu les dades en una carpeta per cada situació que es podria donar a l'habitació, les numera i les agrupa totes en una carpeta externa. Aquestes carpetes que corresponen a cada situació que es podria donar en l'habitació de prova, contenen un fitxer de text amb les diferents dades que conformen la situació. En aquesta única versió de prova de que es disposava es va utilitzar per llistar com a possible informació del model Situation totes les dades disponibles que conformaven les diferents situacions disponibles, en un cas real hi podria haver moltes més situacions i llavors s'hauria de fer un estudi pertinent incloent totes aquestes dades diferents dins el modelatge de la situació per poder recollir totes les possibles. Aquestes dades en les situacions de prova tampoc eren en tots els casos valors desxifrables, però en una versió operativa aquestes dades que aportarien els fitxers utilitzats si que serien significatives i és per això que s'han guardat igualment aquestes dades i és informació accessible des de l'aplicació, ja que en un futur s'hi hauria de poder accedir per consultar la informació pertinent a la situació.

Per altra banda, les carpetes concretes de cada situació contenen una carpeta anomenada traces que conté un arxiu **XML** amb tots els valors dels diferents factors de la situació en qüestió. També es van modelar aquestes dades amb el model de Factor, fent que cada un dels reports del fitxer **XML** es convertís en un factor, amb el seu nom, valor i posterior anàlisi de les dades per calcular altres dels valors que conformen un factor.

Per a la lectura d'aquests elements locals, es van haver de definir tots els arxius a consultar com a Embedded Resources, tal i com s'ha explicat amb anterioritat. Per a llegir els arxius de text es va utilitzar els `StreamReader` que permetia llegir els arxius amb un **encoding** concret, tot i que l'anàlisi de dades s'havia de dur a terme fent un estudi de l'estructura dels fitxers en qüestió. Pel que feia els **XML**, es van utilitzar els objectes `XDocument` que representa un arxiu **XML** i un cop carregat et permet accedir a les dades que es vulguin.

Pel que fa als hotels que formarien part del sistema d'hotels adherits a l'aplicació, es va crear un arxiu **JSON** que emmagatzema les dades de diferents hotels inventats per poder fer proves, i que l'usuari pogués iniciar sessió en alguna de les habitacions d'algun dels hotels disponibles. A continuació es pot veure l'estructura muntada d'aquest fitxer que actuaria com a base de dades dels diferents hotels que formen part del sistema:

```
    ],  
    "Hotel2": [  
      {  
        "CompanyName": "Hotel A6",  
        "IdHotel": "A6",  
        "City": "Barcelona",  
        "Street": "Campus Nord",  
        "Number": "s/n",  
        "Rooms": [  
          {  
            "IdRoom": "202",  
            "Description": "L'aula 202 és una sala petita situada a la segona planta de l'edifici A6 de la FIB."  
          },  
          {  
            "IdRoom": "203",  
            "Description": "L'aula 204 és una sala gran a la segona planta de l'edifici A6 de la FIB."  
          },  
          {  
            "IdRoom": "204",  
            "Description": "L'aula 204 és una sala situada a la segona planta de l'edifici A6 de la FIB."  
          }  
        ]  
      }  
    ],  
    "Hotel3": [  
      {  
        "CompanyName": "Hotel A7",  
        "IdHotel": "A7",  
        "City": "Barcelona",  
        "Street": "Campus Nord",  
        "Number": "s/n",  
        "Rooms": [  
          {  
            "IdRoom": "205",  
            "Description": "L'aula 205 és una sala petita situada a la segona planta de l'edifici A7 de la FIB."  
          },  
          {  
            "IdRoom": "206",  
            "Description": "L'aula 206 és una sala gran a la segona planta de l'edifici A7 de la FIB."  
          },  
          {  
            "IdRoom": "207",  
            "Description": "L'aula 207 és una sala situada a la segona planta de l'edifici A7 de la FIB."  
          }  
        ]  
      }  
    ]  
  ]  
}
```

Figura 29: Un dels hotels que conforma el hotels.json

Per llegir aquests fitxers es va utilitzar la llibreria JSON.NET que permetia deserialitzar directament els elements del JSON sempre i quan tinguessin els mateixos noms i estructura que els models en que es volia deserialitzar.

No només es va utilitzar per els hotels aquest sistema de serialització dels JSON, sinó que també es va utilitzar per a llegir la clau del client. La idea és que els diferents hotels que integrarien el sistema, atorgarien als clients entrants una clau virtual (un JSON) que actuaria com a comprovació de que aquell usuari realment té permís per accedir a la informació de l'habitació. A l'usuari se li informaria la clau d'entrada i la contrasenya, que al coincidir amb el fitxer de clau que tindria, permetria iniciar sessió i poder accedir a tota la informació de l'habitació on estaria allotjat. La clau de l'usuari seria el identificador de l'hotel seguit de l'identificador de l'habitació. Ara per ara doncs la clau virtual està localment integrada al projecte per mostrar com es fa la comprovació de les dades que entren des del formulari d'iniciar sessió amb el fitxer en qüestió. Els hotels tindrien un sistema per a generar contrasenyes aleatòries pels usuaris, i cada habitació del seu hotel contindria un identificador únic i l'hotel tindria un identificador únic de mínim 2 caràcters i aquest identificador sencer no podria ser el començament de cap altra identificador d'hotel. Aquesta comprovació es faria amb el sistema de generació de claus pels hotels, aquestes restriccions hi serien perquè no hi pogués haver col·lisió amb la cerca de l'existència de la clau d'usuari a l'hora d'iniciar sessió.

Finalment també cal comentar que a l'utilitzar el Settings, que s'ha explicat anteriorment, pot portar a la confusió i semblar que s'estan emmagatzemant les dades de l'usuari que ha iniciat sessió en una base de dades, però el cas és que simplement es guarden les dades necessàries gràcies a aquesta llibreria que permet guardar elements com si es tractés d'una sessió.

La informació rellevant que s'hi guarda en aquest fitxer és si l'usuari està amb la sessió iniciada o no, el identificador d'aquest usuari en qüestió, i si l'usuari en qüestió ha aconseguit més d'un 90% de estalvi total i se l'ha felicitat.

6. Planificació temporal

6.1. Descripció de les tasques

6.1.1. Treball inicial

En aquesta part es va fer un estudi a fons del projecte juntament amb la seva planificació temporal i els anàlisis pertinents.

6.1.2. Instal·lació i familiarització dels entorns

Tot el projecte es va dur a terme amb un portàtil, i per tant primer de tot va caldre aprendre a treballar amb tots els entorns necessaris en aquest. Primer de tot es va instal·lar Visual Studio ja que mai abans s'havia treballat en cap projecte amb aquestes eines de desenvolupament de Microsoft. També es va treballar amb Xamarin, que permetia treballar fàcilment per tenir un producte desenvolupat per Android, iOS i Windows, tot i que aquest últim no es va seleccionar per la feina addicional que hauria suposat i la poca quota de mercat que té en comparació amb els altres dos sistemes operatius que gaire bé suposen la totalitat d'aquesta [54]. De fet Microsoft mateix ha començat a produir els seus dispositius amb el sistema operatiu Android enlloc del seu Windows. [55][56]

A banda de tot això, es va aprendre el funcionament de **Teamfoundation** que es va utilitzar per compartir el codi amb el director del projecte, tot i que principalment es va utilitzar com a control de versions.

Com que normalment la instal·lació adequada del software porta problemes per tenir totes les versions necessàries i sense problemes a l'executar, es va fer una estimació temporal pessimista d'aquest apartat.

6.1.3. Familiarització amb Xamarin

La feina principal va ser indagar en el tema i aprendre'n tant com va ser possible durant els dies dedicats a l'estudi d'aquest software.

En la planificació inicial aquest apartat també s'hi tenia en compte la familiarització amb OPC-UA però degut a l'abast que es va definir finalment al projecte no va ser necessari investigar fins a tal punt. També era el cas de l'adaptació a les característiques de NECADA, que finalment simplement es van aprendre les característiques bàsiques i com extreure'n la informació necessària i tot això es va fer durant el desenvolupament del projecte.

Sempre que va aparèixer un dubte de Xamarin durant el desenvolupament del projecte es va consultar sense afegir el temps addicional a aquest apartat, sinó al desenvolupament en si mateix.

6.1.4. Desenvolupament de l'aplicació .NET

Aquesta part va ser la que s'hi va dedicar més temps i es va tractar d'aplicar tots els coneixements adquirits en els punts anteriors. Un cop es van conèixer totes les eines va ser l'hora d'aplicar-les tal i com tocava. Així doncs, durant 21 setmanes es van fer 20 sprints de Scrum per tal de generar l'aplicació, tot fet per una mateixa persona adoptant els diferents rols com s'explica més endavant. Setmana a setmana es van fer les planificacions adients per repartir la càrrega de treball entre totes les iteracions.

Inicialment estava pensat que fossin 8 setmanes amb 8 sprints però com explicarem més endavant es va canviar el termini d'entrega i per tant es van ampliar el nombre de setmanes.

6.1.5. Documentar el treball

Durant tot el temps de desenvolupament qualsevol avanç o procés important va ser descrit i afegit a la documentació que s'havia generat en la primera fase del treball, a GEP. Així va ser més fàcil tenir un treball complet i el més acurat possible a tota la informació que s'havia recollit i era necessària presentar.

6.1.6. Revisió i entrega de la versió final

En aquest apartat es va revisar tota la documentació, tan la claredat com els errors ortogràfics que hi podien haver, o també de puntuació o gramaticals. A més a més, es va fer una feina molt important de veure si el document s'adequava a la documentació necessària pel tipus de treball que era. Qualsevol part que sobrés també se'n va fer cribratge durant aquesta fase.

6.2. Planificació inicial

6.2.1 Càrrega de treball de les tasques

A la taula 19 es poden veure una aproximació de les hores que es van dedicar a cada tasca. El treball inicial no apareixer ja es va analitzar a la fita inicial, i els dies que va durar apareix al Diagrama de Gantt inicial.

Tasca	Duració estimada (hores)
Instal·lació dels entorns	65h
Aprenentatge de OPC-UA	15h
Aprenentatge de Xamarin	30h
Adaptació a NECADA	15h
Desenvolupament aplicació	305h

Documentació	45h
Total	475h

Taula 19: Total d'hores del treball per tasques inicial

6.2.2 Diagrama de GANTT

En aquesta secció es poden veure les diferents tasques amb les seves dates assignades i les precedències concretes.

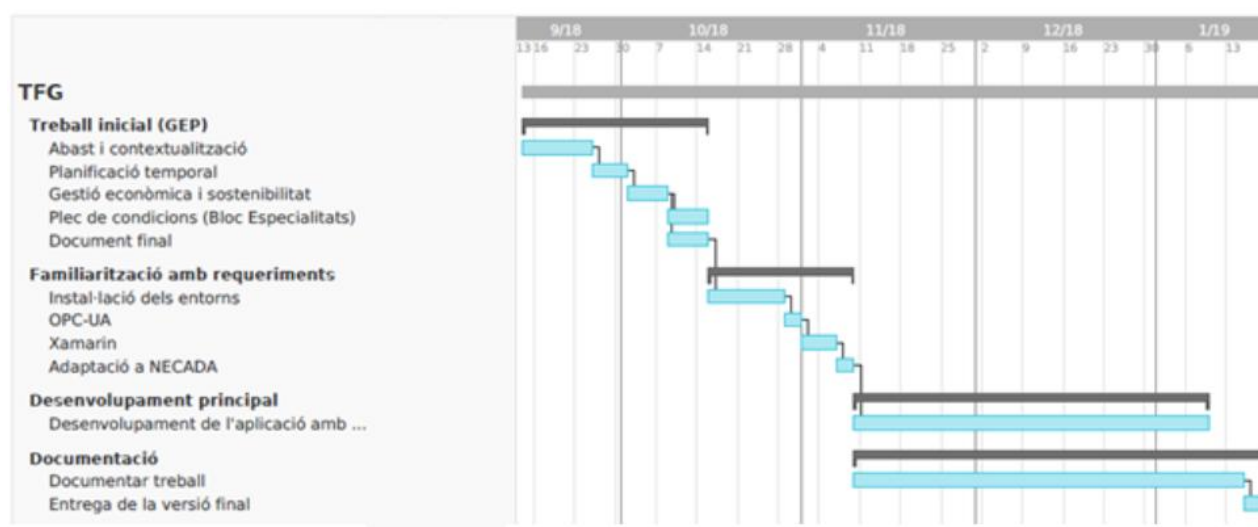


Figura 30: Diagrama de Gantt inicial del projecte creat amb TeamGantt

(*) Les reunions no s'han inclòs al diagrama ja que com s'ha explicat només es van efectuar de forma presencial en cas d'haver-hi problemes molt rellevants, però normalment el seguiment es va fer telemàticament per Skype.

6.2.3. Valoració d'alternatives i pla d'acció

El pla d'acció està definit a l'anterior diagrama de Gantt però no estava clar que aquest planning es pogués seguir d'aquesta manera, ja que treballant amb entorns nous és molt fàcil que apareguin problemes per arribar al resultat final en el temps definit. Tot seguit es llisten quins poden ser aquests problemes i també es valoren les alternatives que es podien adoptar:

6.2.3.1. Planificació

Un error que podia sorgir era la planificació, que es desviés del que s'esperava i llavors s'hagués de reorganitzar i adaptar les metodologies de treball. Es va estar en contacte constantment amb el director del projecte mitjançant Skype on en tot moment va estar al corrent dels avenços i es va poder dur una comunicació fluida. En cas de no arribar, es va preveure apujar 5 hores setmanals en els 4 últims sprints per tal d'intentar arribar correctament a la finalització del projecte.

6.2.3.2. Bugs

Al treballar amb un entorn completament nou com és NECADA i amb llenguatges desconeguts de bon principi com per exemple C#, era molt possible que apareguessin bugs i errors en els diferents codis que es generessin, però es va plantejar solucionar-ho amb una sèrie de tests en acabar cada iteració per comprovar que el codi seguia funcionant.

6.2.3.3. Errors de configuració

Per poder arribar a desenvolupar l'aplicació per a totes les plataformes, va caldre la instal·lació i adaptació de molts programes i extensions que haurien de funcionar correctament els uns amb els altres. Podien aparèixer diversos errors, tot i que la idea era poder-ho solucionar amb el temps que comprendria el temps destinat a la instal·lació d'aquests ja que se'n va fer una previsió de temps pessimista tenint en compte que podien sorgir diversos problemes.

6.2.3.4. Errors de simulació

Es podia donar el cas de que la simulació que efectua NECADA o Visual Studio no hagués sigut adaptable als aparells que es tenien disponibles i per tant s'hagués hagut de treballar des de punts no tant accessibles i còmodes i això dificultés el progrés. Podria ser per exemple el cas de les simulacions de l'aplicació iOS, que Apple només permet fer proves en dispositius seus i no se'n disposava cap, tot i que la idea era informar-se més i solucionar-ho buscant companys que poguessin facilitar els dispositius necessaris.

6.3. Desviació soferta durant el projecte

6.3.1 Canvis respecte les tasques

6.3.1.1. Familiarització amb OPC-UA

Temps després d'haver presentat el projecte final de GEP i per tant d'haver definit les tasques que havia de dur a terme per arribar a la finalització del projecte, es va veure que quedava fora de l'abast la implementació de OPC-UA dins el projecte ja que el que es faria seria un prototip del què podria arribar a ser aquesta aplicació futura i per tant no caldria la comunicació a temps real amb sensors i ja no tenia sentit dominar el llenguatge d'intercanvi d'informació.

6.3.1.2. Adaptació a les característiques de NECADA

Després d'haver fet planificació inicial amb el director es va parlar i concloure que no calia que s'aprengués com funciona NECADA fins el fons de la qüestió, que simplement sabent com es podien tractar les dades que generava ja n'hi havia suficient. Així doncs, va ser hora de centrar-se en saber com tractar-les, però tot això va constar de la part de desenvolupament de l'aplicació, i en cap cas com una part deslligada i a part. Així doncs, aquesta tasca va

deixar de tenir sentit com a una unitat separada i es va fer l'anàlisi pertinent durant el desenvolupament del projecte.

6.3.2 Canvis de termini

Com s'ha comentat amb anterioritat, la idea principal era entregar el projecte abans i això ha generat unes certes desviacions pel que fa la planificació i repercutint en altres aspectes com l'econòmic (com es veurà més endavant), però més enllà d'això mai ha suposat un problema per definir l'abast del projecte sinó que si aquest ha canviat ha estat perquè així es va decidir amb el director segons el que interessava més tractar.

6.3.3 Proves amb iOS

Com s'havia previst de bon inici van ocórrer problemes per poder provar el software generat per iOS. Aquestes versions de l'aplicació només es poden provar amb un MAC a la disposició del desenvolupador i no existeix cap eina que permeti fer la simulació des d'un ordinador Windows sense connectar-se a un Mac.

Així doncs, per poder provar l'aplicació en aquest sistema operatiu es va haver d'aconseguir un Mac d'un company que sempre que ell no el necessitava, i era necessari per comprovar que el desenvolupament funcionava correctament, aquest el deixava sense cap problema. Tot i això, aquest tema va suposar certa incomoditat per el correcte desenvolupament de l'aplicació ja que quan s'havien de comprovar certes coses en diferents iteracions, no es podia perquè no es disposava del dispositiu i s'havia de provar més endavant quan se'n podia disposar. També es va aconseguir un iPhone per poder provar l'aplicació sense fer-ho des d'un simulador, sempre executant-la des de Mac.

A més a més, per connectar els dos dispositius existeix una eina de Visual Studio que permet connectar els dos dispositius si estan connectats a la mateixa xarxa Wi-Fi, però com posteriorment es va trobar pels fòrums de Xamarin hi havia un problema en el sistema de connexió entre dispositius amb la versió que tenia de Visual Studio (2017) instal·lada al meu Windows. Els mateixos desenvolupadors del programari de desenvolupament van solucionar aquests errors arrel d'aquestes discussions dels fòrums i en la següent versió del IDE, la 2019, la van poder treure sense aquest problema [57]. Per solucionar aquest tema, finalment es va optar per simplement connectar-se al repositori creat al **Teamfoundation** des de l'ordinador d'Apple i baixar-se el codi del projecte per poder-lo provar.

Després de tota aquesta problemàtica associada connexió amb el Mac, els problemes que es van mantenir van ser poder seguir disposant del ordinador quan es necessitava que en algun cas potser va fer perdre cert temps entre iteracions de desenvolupament però no molt significativament.

6.4. Planificació final

6.4.1 Càrrega de treball de les tasques

Degut a aquests canvis de tasques, hi va haver hores que s'havien d'haver dedicat a les tasques en qüestió que no es van haver de fer. Tot i que sí que és veritat que hi va haver feines d'adaptació a NECADA durant el desenvolupament de l'aplicació que no es compta enlloc i per tant això també ha suposat un temps addicional.

També hi va haver tasques com la instal·lació que va ser costosa però no tant com s'esperava en el cas pessimista. Tot i això, un cop es va saber que es podria entregar el treball més endavant i es va veure que amb el temps setmanal que s'hi podia dedicar era inviable arribar a l'entrega, es pot veure com les hores de dedicació per dia van baixar i això va fer que algunes tasques es triguessin més dies a implementar dels esperats i tot el projecte es relaxés més.

A la taula 20 es pot veure una aproximació de les hores que s'han dedicat a cada tasca. Les dedicades al treball inicial de GEP no apareixen:

Tasca	Duració estimada (hores)
Instal·lació dels entorns	55h
Aprenentatge de Xamarin	90h
Desenvolupament aplicació	380h
Documentació	50h
Total	575h

Taula 20: Total d'hores del treball per tasques final

6.4.2 Diagrama de GANTT

Tal i com s'ha explicat en l'apartat anterior, d'un bon inici s'esperava que el treball tingués una durada més curta però finalment per circumstàncies alienes del treball es va triar fer la pròrroga de la presentació del treball.

Tenint en compte el que s'ha esmentat i que algunes tasques que a l'inici preveien fer i finalment no va ser el cas, així és com va quedar finalment el diagrama:

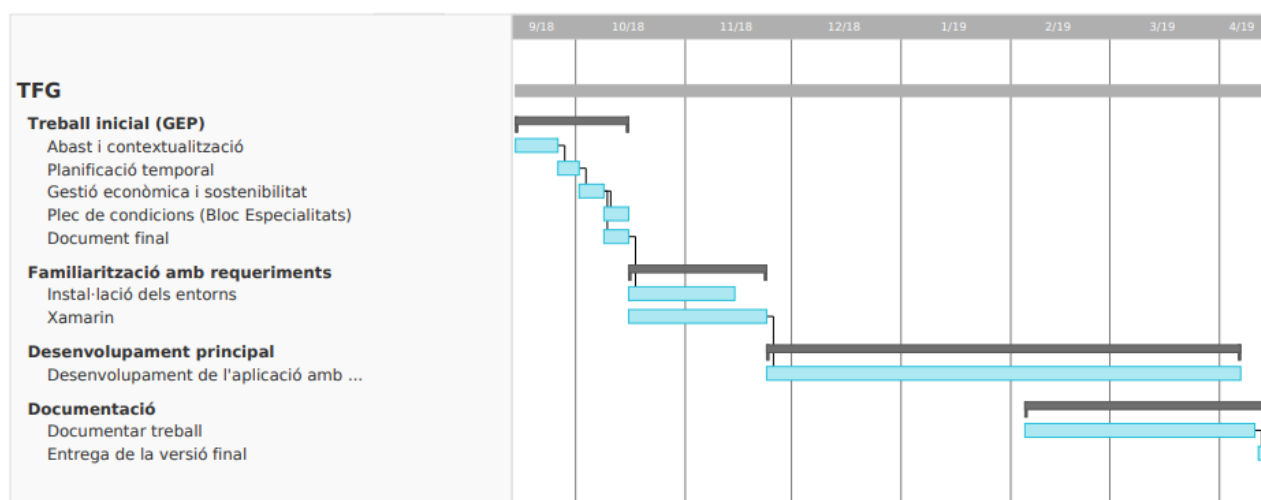


Figura 31: Diagrama de Gantt final del projecte creat amb TeamGantt

7. Gestió econòmica

Aquest apartat del treball s'encarrega de quantificar el cost que va suposar dur a terme el projecte. Per poder-ho fer es van tenir en compte les diferents tasques que calia fer que s'han identificat prèviament, i tot el material necessari que va permetre dur-ho a terme. Primerament els càlculs que es van fer comptaven que el projecte es faria amb quatre mesos, per tant les taules que tenen sentit estar desdoblades segons la previsió inicial i la, i així es pot veure el canvi que va ocórrer.

7.1. Costos directes

7.1.1. Costos en hardware

En la següent taula es pot veure els preus de tot els productes físics necessaris per tirar endavant el projecte. Se'n fa el càlcul per saber les hores reals que s'hi dedica amb el dispositiu per dur a terme el projecte:

	Hardware	Preu	Unitats	Vida útil	Amortització
	Portàtil	700€	1	6 anys	30,36€
	Mòbil	180€	1	4 anys	11,71€
Total	-	-	-	-	42,07€

Taula 21: Costos directes (Hardware)

Suposicions:

- La vida útil dels portàtils és de 6-8 anys. El que es té és fa 6 anys i segueix funcionant tot i que cada cop es veu més obsolet, així doncs se li donarà aproximadament aquesta esperança de vida útil.
- Vida útil del telèfon mòbil solen tenir una vida útil molt curta, tot i que de normal s'intenta estirar tant com es pugui sempre. Tot i que més de 5 anys és difícil que pugui aguantar als cops i canvis que apareixen.

Observacions:

- També es va disposar d'un Mac i un iPhone, però no ha estat afegit a la taula ja que el seu ús va ser molt puntual i van ser dos aparells que es van aconseguir gràcies a gent propera que els va deixar durant uns dies, però l'ús va ser ínfim com per tenir certa representació al total del treball.

7.1.2. Costos en software

En la següent taula es pot veure el cost de tot el programari necessari per poder crear l'aplicació, que per sort és tot nul. Durant la tasca d'instal·lació d'entorns es va posar a lloc tot el següent programari:

	Software	Preu	Unitats	Vida útil	Amortització
	Windows 7 Professional	0€	1	3	0€
	NECADA	0€	1	3	0€
	Xamarin	0€	1	3	0€
	Teamfoundation	0€	1	3	0€
	Visual Studio Enterprise	0€	1	3	0€
Total	-	-	-	-	0€

Taula 22: Costos directes (Software)

Aclariments:

- Normalment el sistema operatiu de Microsoft i Visual Studio tenen un preu, però els que es tenen instal·lats al portàtil es van instal·lar amb la versió gratuïta que ofereix la universitat.

7.1.3. Costos en recursos humans

En la següent taula s'analitzen els costos humans que suposa el projecte. El cost es fa adequant-se a les hores de cada rol necessàries, sumant el total de les 475h del projecte. Tots aquests rols els efectuarà la mateixa persona i per tant les hores no podran ser fetes simultàniament:

	Rol	Nombre	Preu / hora	Hores / setmana	Estimació de setmanes	Cost inicial
	Cap de projecte	1	28€/h [58]	5h/set	15	2100€
	Programador júnior	1	21€/h [59]	35h/set	11	8085€
	Tester	1	23€/h [60]	5h/set	3	345€
Total	-	-	-	-	-	10530€

Taula 23: Costos directes inicials (Recursos humans)

El cap de projecte es va dedicar plenament a la documentació del projecte i la seva revisió. Per la banda del programador totes les hores van ser dedicades a aprendre els llenguatges necessaris, instal·lar el programari, i desenvolupar l'aplicació. I a cada iteració el tester va fer les proves de test per la release.

El projecte finalment però, va durar 3 mesos més dels previstos, tot i que també es van fer menys hores de les esperades inicialment per setmana, el total van ser 575h:

	Rol	Nombre	Preu / hora	Hores / setmana	Estimació de setmanes	Cost final
	Cap de projecte	1	28€/h [58]	3h/set	26	2184€
	Programador júnior	1	21€/h [59]	20h/set	24	10080€
	Tester	1	23€/h [60]	1h/set	17	391€
Total	-	-	-	-	-	12655€

Taula 24: Costos directes finals (Recursos humans)

Un dels errors que es va fer en la planificació inicial era que el tester només provava al final, però a l'aplicar una metodologia àgil el tester cada setmana havia de fer els testos dels nous avenços del programador, tot i que no va estar el cas de totes perquè no a tots els sprints hi havia una release. Per això hi ha menys setmanes de tester que de programador júnior.

7.1.4. Costos directes totals

	Tipus de cost	Cost inicial	Cost final
	Hardware	42,07€	42,07€
	Software	0€	0€
	Recursos humans	10530€	12655€
Total	-	10572,07€	12697,07€

Taula 25: Costos directes

7.2. Costos indirectes

Els costos indirectes principals que hi va haver en aquest projecte van ser l'electricitat i la tarifa d'Internet.

7.2.1. Electricitat

El preu del kWh a Espanya en el moment de l'estudi estava: 0,13€. [61]

L'energia que consumeix el portàtil utilitzat és de 102,84 W cada hora quan el carregador està connectat, i com que la bateria està malmesa serà el consum d'energia cada hora. Totes les hores que es van dedicar a les tasques del projecte es duïen a terme amb el portàtil. Per tant van ser 0,10284 kWh que amb un total de 475h (inicials) i 575h (finals) de projecte és l'equivalent a: 48,849 kWh i 59,133 kWh respectivament.

Amb el consum de 48,849 kWh / 59,133 kWh i assumint un preu de 0,13 €/kWh queda un total de:

Inicial: 6,35037 € ~ Total: 6,35€

Final: 7,68729 € ~ Total: 7,67€

7.2.2. Internet

Per la seva banda, el servei d'Internet que es tenia contractat en el moment de l'estudi costava un total de 45 € al mes. El servei realment es compartia amb tres companys, però per simplificar el càlcul, ja que també es repartien els MB contractats, comptarem amb la totalitat de la despesa.

Per tant, en la planificació inicial s'estava parlant de 45 €/mes durant 4 mesos, si cada dia comptàvem treballar 5h de les 8h que seria normal:

$$45 \text{ €/mes} * 4 \text{ mesos} = 180\text{€} * 5\text{h}/8\text{h} = 112,5\text{€}$$

Pel que fa la final, van ser 45€/mes durant 7 mesos, amb cada dia treballant unes 4h de les 8h que seria normal:

$$45\text{€/mes} * 7 \text{ mesos} = 315\text{€} * 4\text{h}/8\text{h} = 157,5\text{€}$$

7.2.3. Costos indirectes totals

	Tipus de cost indirecte	Cost total inicial	Cost total final
	Electricitat	6,35 €	7,67 €
	Internet	112,5€	157,5€

Total	-	118,85€	165,17€
--------------	---	----------------	----------------

Taula 26: Costos indirectes

7.3. Costos directes i indirectes

	Tipus de cost	Cost inicial	Cost final
	Directe	10572,07€	12697,07€
	Indirecte	118,85€	165,17€
Total	-	10690,92€	12862,24€

Taula 27: Sumatori de costos directes i indirectes

7.4. Costos de contingència

Com que la planificació va ser molt optimista en algunes seccions no es creu que hi pugui haver gaire desviament, així doncs el percentatge de contingència serà menor, i aquí hi ha el càlcul que es va fer amb els costos directes i indirectes:

	Explicació	Impacte al cost	Cost total inicial	Cost total final
	Nivell de detall és bo	6%	641,45€	771,73€
Total	-	-	641,45€	771,73€

Taula 28: Costos de contingència

7.5. Costos d'imprevistos

Tal i com s'ha comentat al llarg de la documentació, és possible que el projecte hagués requerit que s'haguessin implicat més hores en algun moment. Si no s'hagués arribat a acabar el projecte els últims sprints s'hauria pujat un total de 5h per setmana per arribar:

	Causes	Solució	Risc	Impacte al cost	Cost inicial	Cost final
	Falta de temps al desenvolupament de l'aplicació	Pujar de 35 a 40 hores als últims 4 sprints del programador	25%	500€	125€	0€
Total	-	-	-	-	125€	0€

Taula 29: Costos d'imprevistos

No s'adjuntarà la versió final ja que no va ser necessari fer l'augment d'hores.

7.6. Costos totals i control de gestió

Finalment doncs, es tenen els costos totals de crear el projecte:

	Tipus de cost	Cost inicial	Cost final
	Directe i indirecte	10690,92€	12862,24€
	Contingència	641,45€	771,73€
	Imprevistos	125€	0€
Total sense IVA	-	11457,37€	13633,97€
Total amb 21% de IVA	-	13863,41€	16497,10€

Taula 30: Cost total amb IVA i sense

Com es pot veure al pressupost no hi va afectar molt que poguessin haver sorgit imprevistos i és per això que no es va tenir una política molt concreta en cas d'ocórrer. Tot i així, calia ser conscient que si es donava el cas el motiu seria perquè el desenvolupament de l'aplicació s'hauria encallat en algun punt i per tant la solució passaria per augmentar les hores de desenvolupament de l'aplicació.

8. Identificació de lleis i regulacions

La gran majoria de software que es pot integrar d'altres autors ve regulat per una sèrie de normes a seguir que han escollit els seus creadors.

En aquest projecte es van utilitzar diferents llibreries totes publicades a NuGet: un sistema de gestió de paquets gratuït i de codi obert que està integrat al **IDE** que utilitzem per desenvolupar el projecte. Tots aquests paquets que es poden obtenir de forma gratuïta tenen un codi accessible (majoritàriament a través de **GitHub**).

La llicència que té cada paquet publicat a NuGet està subjecte als termes i condicions propis del paquet, no existeix quelcom genèric que estigui regit per la plataforma distribuïdora. [62]

Així doncs es llisten una per una les llibreries utilitzades, que se n'ha explicat les funcions específiques al punt 3.5, comentant-ne les llicències que se'ls aplica:

- **FlowListView** [63]: Té la **Llicència Apache 2.0**, de software lliure, que dona llibertat d'ús i cap garantia, i sobretot i com a principal condició requereix l'avís dels drets d'autor (copyright). Com a usuari d'aquest software permet la llibertat d'utilitzar-lo per qualsevol propòsit. Tot i que està regulat quan entren en joc modificacions del codi: Obliga a llistar explícitament totes les modificacions que es fan del software original. També especifica clarament que de cap manera es pot anomenar al teu producte amagant que el producte està avalat per Apache. En aquest cas no s'ha fet cap modificació del codi simplement s'ha utilitzat el codi, per tant, només cal que mencionar-ne l'autoria. [64]
- **PullToRefresh** [65]: Té la **Llicència MIT** que és una llicència de programari lliure permissiva que només s'encarrega de preservar el copyright i permet utilitzar, copiar, modificar o integrar-ho amb un altre software sense cap mena de restricció. És de les llicències més permissives, només cal assegurar-se d'afegir una còpia de la llicència original MIT i el copyright per informar-ne. Per tant aquest tipus de llicència no genera cap problema en poder-la utilitzar lliurement pel projecte sempre que se'n mencioni l'autoria.
- **Settings** [66]: Té la **Llicència MIT** que ha estat explícitament detallada en la explicació de PullToRefresh d'aquest mateix llistat.
- **JSON.NET** [42]: Té la **Llicència MIT** que ha estat explícitament detallada en la explicació de PullToRefresh d'aquest mateix llistat.

A un dels fitxers principals de l'aplicació, el App.xaml.cs, es van mencionar totes les llicències seguint així les restriccions que aquestes reglamentaven.

A part de les diferents llibreries que es van utilitzar, el projecte **NECADA** [67] del qual se n'extreuen dades (de l'edifici de proves predeterminat) es van utilitzar amb el consentiment de l'equip creador del projecte. És un treball nascut a la UPC el responsable del qual n'és el

Pau Fonseca, el director del projecte, que és qui va atorgar aquests drets per la utilització de la informació d'aquest edifici de prova.

Les fons que s'han utilitzat pel contingut de l'aplicació són Gill Sans MT i les seves versions en negreta (Gill Sans MT Bold) i la versió estreta (Gill Sans MT Condensed). Aquestes fons són propietat de Monotype [68] i en alguns dispositius hi són disponibles aquestes fons, però per assegurar-se de la seva correcta visualització es van incloure al projecte. El problema és que no són fons lliures [69], però per tenir un disseny com es volia en el prototip es va pensar que era millor mostrar-la tal i com s'havia imaginat i en un cas futur de comercialització de l'aplicació ja es farien els tràmits necessaris perquè l'ús d'aquestes fos adequat. La descàrrega es va fer des de dafontFree una pàgina que disposa de moltes fons diferents que es poden descarregar gratuïtament. Molts cops aquestes fons són incompletes o versions de prova, però en aquest cas les fons descarregades van complir l'objectiu que era poder mostrar la interfície tal i com s'havia imaginat sense cap problema.

Les icones que es van utilitzar van ser dissenyades amb la col·laboració d'una altra persona, la Mar Guinart, i hi ha permís total per utilitzar-los en aquest projecte ja que van ser dissenyades per aquest fi. Només n'hi ha dues extrems d'Internet, i aquestes són la icona que apareix en la vista del Perfil de l'usuari i la de tancar la sessió. Es van extreure d'una pàgina que proporciona icones de manera gratuïta per projectes no comercials [70], i per tant s'haurien de poder utilitzar lliurement. Tot i això, aquesta pàgina no assegura que tots els seus continguts no tinguin llicència, però en aquest cas estaria mal regulat des de la plataforma d'on s'ha obtingut. Se'n podria estar incomplint la normativa, ja que no es pot saber la llicència de les imatges originals, però no hi ha d'haver cap problema sense ser un projecte comercialitzat i havent-les modificat fins i tot de color.

9. Informe de Sostenibilitat

En aquest informe es mesurarà l'impacte sobre la sostenibilitat que genera aquest projecte. Es farà des de tres òptiques diferents: des de l'impacte social, l'ambiental i l'econòmic.

9.1. Dimensió econòmica

Anteriorment s'ha analitzat el cost que costaria el projecte per posar-lo a producció (PPP), i analitzant el cost del personal necessari, tots els costos de hardware i software juntament amb l'electricitat i Internet que necessitarem i aplicant costos de contingència, possibles imprevistos i l'IVA la suma total és de: 16497,10€.

Pot semblar un cost molt elevat i impossible de plantejar per un estudiant sense cap mena d'ajut, però tal i com s'havia dit a l'apartat 4.1.3., un possible interessat d'aquest projecte és el Departament de Territori i Sostenibilitat. Si aquest departament hi estigués interessat i hi volgués invertir, i fins i tot ajudar a trobar inversors, podria ser un cost molt assumible. Tot i que s'ha de tenir en compte que la major part del cost prové de les hores de treball que s'haurien de pagar i en aquest cas les hores aportades seran pel Treball de Final de Grau i no hi hauria cap mena de pagament i per tant es reduirien 12655€ de costos de Recursos Humans i per tant seria més que assumible. A més a més, aquestes hores dedicades realment podrien ser menors si es dugués a terme per un equip on les tasques es poguessin fer de manera simultània quan correspongués. Cal tenir en compte però que aquests costos i càlculs són pel prototip que s'ha creat amb aquest treball, i per tant per la versió definitiva en que podria estar interessat el Departament caldrien més hores per acabar d'ajustar la connexió amb NECADA.

Es va tenir en compte en aquesta aproximació de costos l'estalvi de codi mitjançant llibreries de codi, i tot això fa que acabi suposant una despesa menor de la que podria acabar suposant en cas de no utilitzar-ne.

Un cop finalitzat, és el sector hotelier, els hotels, qui decidiran si voldrien implementar aquesta millora a la seva proposta d'establiment. Aquests haurien de fer el correcte manteniment dels sensors, i pagar un preu per poder tenir el sistema, i com més l'utilitzessin els seus clients això els hi aportaria uns beneficis a l'estalviar energia. Amb aquest preu que els hotels haurien de pagar, es podria fer viable tot el pagament de servidors i infraestructures que fossin necessàries per poder mantenir actius els sistemes d'anàlisi de cada hotel i a més a més treure'n benefici. Tot i que possibles actualitzacions també suposarien uns certs costos que s'haurien de tenir en compte.

9.2. Dimensió ambiental

Tal i com anteriorment es va analitzar en l'apartat 7.2.1 el cost total d'energia durant el desenvolupament de l'aplicació i tots els passos previs van tenir un consum d'un total de 48,849 kWh. Aquesta quantitat d'energia suposa una emissió de 18,07kg de CO₂ [71].

Aquest va ser el principal malbaratament durant la producció ja que tot el treball es feia des d'aparells electrònics que van estar calculats prèviament amb la xifra donada. Tot i això, es

va intentar sempre que es podia utilitzar llibreries i utilitzar el codi compartit per minimitzar les hores de programació i per tant d'utilització d'electricitat per el projecte.

Durant la seva vida útil el projecte consumirà l'energia dels servidors de NECADA, treballant per aportar les dades a cada sistema on s'implementi: però no serà un consum tan important com per tenir-lo en compte.

Tot i aquest malbaratament, com que l'aplicació té una finalitat de crear un ambient sostenible com a principal preocupació, finalment acabarà tenint més impacte positiu que aparentment sembla que pugui tenir. Per tant la empremta ecològica disminuirà gràcies a la creació d'aquest projecte a base de que els usuaris cada vegades l'utilitzin més.

Per tant, si aquest projecte no existís tot el que aconseguix estalviar el sistema NECADA connectat amb l'aplicació seria el consum i impacte ambiental que no es reduiria. Per tant no hi ha cap mena de risc de què el projecte tingui un impacte ambiental negatiu, sinó que en cas de no implementar-se tot seguirà com està però si s'implementa la situació ambiental anirà a millor.

9.3. Dimensió social

Aquest projecte suposa barrejar dues coses molt presents a l'actualitat que són la tecnologia i la preservació del medi ambient. També suposa endinsar-se en un món que és necessari millorar i que cal ajudar a que es cuidi més. A més, en un planeta que li és urgent una intervenció i cada cop la gent n'és més conscient, el sistema podria encaixar a part de tots els avantatges que porta a l'usuari i sector hotelier.

També fer un treball així suposa un canvi en l'entorn ja que informant-se de moltes coses per realitzar-lo, significa generar debat amb companys i amics mostrant dades interessants com pot ser per exemple que els edificis suposen un 40% del consum d'energia total, entre moltes d'altres dades, coses que fan que de mica en mica tothom es vagi conscienciant més d'aquests problemes tant importants.

Un cop es posés en marxa el projecte, només hi hauria avantatges tant com pels creadors, com els consumidors i pel sector hotelier. Tots en la seva mesura rebrien benefici econòmic i per sobre de tot ajuden a no contaminar tant i no malbaratar tanta energia, per tant, la qualitat de vida de les persones es veuria millorada.

Els únics que podrien perdre-hi són les grans elèctriques o empreses energètiques veient-se afectat per la disminució del consum, però seria tan minuciós i els seus beneficis són tan enormes que tampoc els hi afectaria i per tant no es preveu una forta oposició d'aquestes companyies.

10. Conclusions del projecte

Un cop acabat el projecte el que puc dir és que dur a terme aquest treball ha suposat una gran experiència poder aprendre des de zero diferents llenguatges i una nova metodologia de crear aplicacions i el resultat que se n'ha aconseguit m'ha satisfet molt.

Potser va canviar una mica l'abast del projecte ja que a mesura que aquest va anar avançant conjuntament amb el director anàvem definint fins on podríem arribar. Es tenia clar quins eren els objectius i a partir d'aquí vam anar veient què interessava potenciar més.

Pel que fa el desenvolupament de l'aplicació va ser llarg però en cap cas va ser feixuc. Ha acabat esdevenint una aplicació molt simple i bàsica però crec que a l'hora s'ha aconseguit complir tots els objectius que s'havien proposat. Aquesta simplicitat en cap cas es veu com una cosa dolenta sinó que es veu com una arma potent per satisfer tots els requisits que s'havien proposat.

Quan vaig triar el treball que proposava el meu director ho vaig fer perquè m'interessa molt la vessant verda de la informàtica, ja que masses cops penso que només es pensa en el màxim benefici i masses pocs cops a quin cost. Estic molt content del prototip d'aplicació que m'ha quedat i de fet m'agradaria poder seguir treballant en aquesta línia i en un futur poder acabar de connectar el sistema amb NECADA i poder provar l'aplicació a l'habitació analitzada a temps real.

Aquest projecte ha pretès mostrar que com a informàtics podem utilitzar totes les dades que es recullen de l'entorn que rodeja per posar la informació a la disposició de les persones i poder generar canvis positius en els comportaments.

Crec que és necessari que el món informàtic segueixi fent passos en aquesta direcció ja que tenim a les nostres mans la capacitat de poder capgirar el sistema de consum actual.

11. Referències

- [1] Fonseca i Casas P, Fonseca i Casas A, Garrido-Soriano N, Casanovas J. Formal simulation model to optimize building sustainability. *Adv Eng Softw* [En línia]. 2014 Mar [cited 2014 Jul 21];69:62-74. Disponible a: <http://www.sciencedirect.com/science/article/pii/S0965997813001877>
- [2] Fonseca i Casas P, Fonseca i Casas A. COMBINING FORMAL DEFINITION OF A SIMULATION MODEL WITH HEURISTICS TO IMPROVE. In: Rabe M, Juan AA, Mustafee N, Skoogh A, Jain S, Johansson B, editors. *Proceedings of the 2018 Winter Simulation Conference* [En línia]. 2018. Disponible a: <http://e-journal.uajy.ac.id/14649/1/JURNAL.pdf>
- [3] R. H. Insider Business, «The world wastes more than twice as much energy as it uses every year», *Business Insider*. [En línia]. Disponible a: <https://www.businessinsider.com/most-energy-still-comes-from-oil-2015-10>. [Consulta: 23-set-2018].
- [4] O. US EPA, «Cleaning Up Electronic Waste (E-Waste)», *US EPA*, 18-març-2014. [En línia]. Disponible a: <https://www.epa.gov/international-cooperation/cleaning-electronic-waste-e-waste>. [Consulta: 23-set-2018].
- [5] «Green Computing - Environmental Issues - The Carnegie Cyber Academy - An Online Safety site and Games for Kids». [En línia]. Disponible a: <http://www.carnegiecyberacademy.com/facultyPages/environment/issues.html>. [Consulta: 23-set-2018].
- [6] «About - NECADA project». [En línia]. Disponible a: <https://necada.wordpress.com/about-2/>. [Consulta: 23-set-2018].
- [7] «Los edificios, responsables de más del 40% del consumo de energía», *ELMUNDO*, 04-nov-2016. [En línia]. Disponible a: <http://www.elmundo.es/economia/2016/11/04/581c51c946163f1f4d8b457e.html>. [Consulta: 23-set-2018].
- [8] «How much energy is consumed in U.S. residential and commercial buildings? - FAQ - U.S. Energy Information Administration (EIA)». [En línia]. Disponible a: <https://www.eia.gov/tools/faqs/faq.php?id=86&t=1>. [Consulta: 23-set-2018].
- [9] Page Laura Epler «Green Computing - History, Methodologies, Benefits and Barriers». [En línia]. Disponible a: http://mason.gmu.edu/~pepler/Paige%20Epler_IT103_ResearchPaper.pdf. [Consulta: 23-set-2018].
- [10] «History of Green Technology in Computing: Regulations», *Bright Hub*. [En línia]. Disponible a: <https://www.brighthub.com/environment/green-computing/articles/71176.aspx>. [Consulta: 24-set-2018].
- [11] «Introduction: A background to the study of green computing and scientometrics». [En línia]. Disponible

a:http://shodhganga.inflibnet.ac.in/bitstream/10603/9631/10/10_chapter%201.pdf.
[Consulta: 24-set-2018].

[12] «Green Computing». [En línia]. Disponible a:
<http://ijarcs.info/index.php/ijarcs/article/viewFile/2755/2743>. [Consulta: 24-set-2018].

[13] «Sustainability | Free Full-Text | Analysis of Applications to Improve the Energy Savings in Residential Buildings Based on Systemic Quality Model | HTML». [En línia]. Disponible a:<https://www.mdpi.com/2071-1050/8/10/1051/html>. [Consulta: 24-set-2018].

[14] Antoni Fonseca i Casas, Pau Fonseca i Casas, i Josep Casanovas «Analysis of Applications to Improve the Energy Savings in Residential Buildings Based on Systemic Quality Model». [En línia]. Disponible a:<https://www.mdpi.com/2071-1050/8/10/1051/html>. [Consulta: 29-set-2018].

[15] «Sistema Intel·ligent per la Gestió del Comportament Energètic Eficient d'usuari en Edificis (ACE)». [En línia]. Disponible a:<https://inlab.fib.upc.edu/ca/sistema-intelligent-la-gestio-del-comportament-energetic-eficient-dusuari-en-edificis-ace>. [Consulta: 29-set-2018].

[16] «App Store: Meter Readings». [En línia]. Disponible a:<https://itunes.apple.com/us/app/meter-readings/id320551309?mt=8>. [Consulta: 29-set-2018].

[17] «Utility Tracking - Energy Benchmarking | WegoWise». [En línia]. Disponible a:<https://www.wegowise.com/>. [Consulta: 29-set-2018].

[18] Oriol Ripoll «Què és la gamificació en educació?». [En línia]. Disponible a:https://www.educaciodema.cat/sites/default/files/article_gamificacio_educativa_oriol_ripoll_0.pdf. [Consulta: 24-set-2018].

[19] «Gamificació a l'empresa: aplicacions en recursos humans | Blog de ESEC: Centre formatiu a Sabadell». [En línia]. Disponible a:<http://blog.esec.cat/2017/10/17/gamificacio-empresa-aplicacions-recursos-humans/>[Consulta: 25-set-2018].

[20] «Què és la gamificació? - Aida Jurado», *Via empresa*. [En línia]. Disponible a:https://www.viaempresa.cat/l-expert/que-es-la-gamificacio_4027_102.html. [Consulta: 25-set-2018].

[21] UB «La gamificació». [En línia]. Disponible a:https://www.ub.edu/portal/documents/814711/1160320/Gamificacio_extens_f.pdf/98a9b797-957b-4392-a50e-9d000dc1cf5e. [Consulta: 25-set-2018].

- [22] M. Sailer, J. U. Hense, S. K. Mayr, i H. Mandl, «How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction», *Computers in Human Behavior*, vol. 69, p. 371-380, abr. [En línia]. Disponible a: <https://www.sciencedirect.com/science/article/pii/S074756321630855X> [Consulta: 25-set-2018].
- [23] Jakob Marovt, «Gamification of software applications», capítol 1, introducció. [En línia]. Disponible a: <http://eprints.fri.uni-lj.si/1851/1/Marovt-1.pdf> [Consulta: 25-set-2018].
- [24] «Mobile apps that have been used only once 2018 | Statistic», *Statista*. [En línia]. Disponible a: <https://www.statista.com/statistics/271628/percentage-of-apps-used-once-in-the-us/>. [Consulta: 25-set-2018].
- [25] H. W. Mak, «5 Gamified Environmental Apps For Sustainable Living», *Gamification Co*, 14-oct-2015. [En línia]. Disponible a: <http://www.gamification.co/2015/10/14/5-gamified-environmental-apps-for-sustainable-living/>. [Consulta: 25-set-2018].
- [26] conceptdev, «Xamarin.Forms - Xamarin.» [En línia]. Disponible a: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/>. [Consulta: 12-Apr-2019].
- [27] gewarren, «Creating a UI with XAML Designer - Visual Studio.» [En línia]. Disponible a: <https://docs.microsoft.com/en-us/visualstudio/designers/creating-a-ui-by-using-xaml-designer-in-visual-studio>. [Accessed: 12-Apr-2019].
- [28] «The Good and The Bad of Xamarin Mobile Development», *AltexSoft*. [En línia]. Disponible a: <https://www.altexsoft.com/blog/mobile/pros-and-cons-of-xamarin-vs-native/>. [Consulta: 12-Apr-2019].
- [29] conceptdev, «Guía multiplataforma - Xamarin.» [En línia]. Disponible a: <https://docs.microsoft.com/es-es/xamarin/cross-platform/>. [Consulta: 12-Apr-2019].
- [30] «Xamarin.Forms 3 - What's upcoming? | DotNetCurry.» [En línia]. Disponible a: <https://www.dotnetcurry.com/xamarin/1393/xamarin-forms-v3-new-features>. [Consulta: 12-Apr-2019].
- [31] *How to port desktop applications to .NET Core 3.0.* <https://channel9.msdn.com/Shows/On-NET/How-to-port-desktop-applications-to-NET-Core-30>
- [32] davidbritch, «Images in Xamarin.Forms - Xamarin.» [En línia]. Disponible a: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/user-interface/images>. [Consulta: 12-Apr-2019].

[33] “Model-View-ViewModel (MVVM) Explained,” *Wintellect*, 24-Apr-2014. [En línia]. Disponible a: <https://www.wintellect.com/model-view-viewmodel-mvvm-explained/>. [Consulta: 12-Apr-2019].

[34] davidbritch, “The Model-View-ViewModel Pattern - Xamarin.” [En línia]. Disponible a: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm>. [Consulta: 12-Apr-2019].

[35] “NuGet Gallery | Home.” [En línia]. Disponible a: <https://www.nuget.org/>. [Consulta: 12-Apr-2019].

[36] karann-msft, “NuGet Documentation.” [En línia]. Disponible a: <https://docs.microsoft.com/en-us/nuget/>. [Consulta: 12-Apr-2019].

[37] D. Luberda, *Xamarin.Forms Custom Controls. Contribute to daniel-luberda/DLToolkit.Forms.Controls development by creating an account on GitHub*. 2019. <https://github.com/daniel-luberda/DLToolkit.Forms.Controls/tree/master/FlowListView/>

[38] “[FlowListView] Trying to use it coming from ListView · Issue #240 · daniel-luberda/DLToolkit.Forms.Controls,” *GitHub*. [En línia]. Disponible a: <https://github.com/daniel-luberda/DLToolkit.Forms.Controls/issues/240>. [Consulta: 12-Apr-2019].

[39] J. Montemagno, *Pull To Refresh a ScrollView or ListView in Xamarin.Forms: jamesmontemagno/Xamarin.Forms-PullToRefreshLayout*. 2019. <https://github.com/jamesmontemagno/Xamarin.Forms-PullToRefreshLayout>

[40] “Xam.Plugins.Settings 3.1.1.” [En línia]. Disponible a: <https://www.nuget.org/packages/Xam.Plugins.Settings/>. [Consulta: 12-Apr-2019].

[41] J. Montemagno, *Read and Write Settings Plugin for Xamarin and Windows: jamesmontemagno/SettingsPlugin*. 2019. <https://github.com/jamesmontemagno/SettingsPlugin>

[42] “Newtonsoft.Json 12.0.1.” [En línia]. Disponible a: <https://www.nuget.org/packages/Newtonsoft.Json/>. [Consulta: 22-Mar-2019].

[43] “App Store - Support - Apple Developer.” [En línia]. Disponible a: <https://developer.apple.com/support/app-store/>. [Consulta: 12-Apr-2019].

[44] “Distribution dashboard | Android Developers.” [En línia]. Disponible a: <https://developer.android.com/about/dashboards>. [Consulta: 12-Apr-2019].

[45] “It’s 2018, but the world’s most popular version of Android is from 2015 - BGR.” [En línia]. Disponible a: <https://bgr.com/2018/01/09/android-oreo-features-sorry-nope-enjoy-marshmallow/>. [Consulta: 12-Apr-2019].

- [46] «Optisim - Necada landing page». [En línia]. Disponible a: <https://necada.com/>. [Consulta: 23-set-2018].
- [47] “Volere_Requirements Specification Template 2015.” ER, FIB (UPC). 2015. [Consulta: 20-Mar-2019].
- [48] “Real Decreto 1720/2007, de 21 de diciembre, por el que se aprueba el Reglamento de desarrollo de la Ley Orgánica 15/1999, de 13 de diciembre, de protección de datos de carácter personal.” [En línia]. Disponible a: http://noticias.juridicas.com/base_datos/Admin/rd1720-2007.html. [Consulta: 25-Mar-2019].
- [49] conceptdev, “Sharing Code Overview - Xamarin.” [En línia]. Disponible a: <https://docs.microsoft.com/en-us/xamarin/cross-platform/app-fundamentals/code-sharing>. [Consulta: 23-mar-2019].
- [50] J. Garcia, “Qué es el Flat Design o Diseño Plano,” *Departamento de Internet*, 02-Sep-2013. [En línia]. Disponible a: <https://www.departamentodeinternet.com/que-es-flat-design-diseno-plano/>. [Consulta: 12-Apr-2019].
- [51] “Android Asset Studio.” [En línia]. Disponible a: <http://romannurik.github.io/AndroidAssetStudio/>. [Consulta: 12-Apr-2019].
- [52] T. Codrops, “Condensed fonts: The good, the bad, the ugly,” *Codrops*, 20-Mar-2012. [En línia]. Disponible a: <https://tympanus.net/codrops/2012/03/20/condensed-fonts-the-good-the-bad-the-ugly/>. [Consulta: 12-Apr-2019].
- [53] “Splash Screen and Android load time - Excessive?,” *Xamarin Community Forums*. [En línia]. Disponible a: <https://forums.xamarin.com/discussion/92453/splash-screen-and-android-load-time-excessive>. [Consulta: 12-Apr-2019].
- [54] A. Casas, “iPhone vs Android: cuota de mercado,” *PCWorld*. [En línia]. Disponible a: <https://www.pcworld.es/articulos/smartphones/iphone-vs-android-cuota-de-mercado-3692825/>. [Consulta: 21-mar-2019].
- [55] L. Sacristán, “La nueva Microsoft móvil se construye sobre Android,” *Xataka Android*, 04-Oct-2018. [En línia]. Disponible a: <https://www.xatakandroid.com/sistema-operativo/nueva-microsoft-movil-se-construye-android>. [Consulta: 21-mar-2019].
- [56] J. Pastor, “Windows 10 Mobile dirá adiós definitivamente en diciembre de 2019,” *Xataka*, 18-Jan-2019. [En línia]. Disponible a: <https://www.xataka.com/moviles/windows-10-mobile-dira-adios-definitivamente-diciembre-2019>. [Consulta: 21-mar-2019].

[57] <https://developercommunity.visualstudio.com/content/problem/374945/key-openssh-is-not-supported.html>

[58] «Sueldos en Product manager en España | Indeed.es.» [En línia]. Disponible a: <https://www.indeed.es/salaries/Product-manager-Salaries>. [Consulta: 08-Oct-2018].

[59] «Sueldos en Desarrollador/a junior en España | Indeed.es.» [En línia]. Disponible a: <https://www.indeed.es/salaries/Desarrollador/a-junior-Salaries>. [Consulta: 08-Oct-2018].

[60] «Sueldos en Tester/a en España | Indeed.es.» [En línia]. Disponible a: <https://www.indeed.es/salaries/Tester/a-Salaries>. [Consulta: 08-Oct-2018].

[61] «Consulta el precio de la luz hoy: Detalles y Evolución de la tarifa PVPC | Tarifaluzhora.» [En línia]. Disponible a: <https://tarifaluzhora.es/?tarifa=normal&fecha=2018-10-08>. [Consulta: 08-Oct-2018].

[62] shishirx34, “NuGet Frequently-Asked Questions.” [En línia]. Disponible a: <https://docs.microsoft.com/en-us/nuget/faqs/nuget-faq>. [Consulta: 23-Mar-2019].

[63] D. Luberda, *Xamarin.Forms Custom Controls. Contribute to daniel-luberda/DLToolkit.Forms.Controls development by creating an account on GitHub*. 2019. [En línia]. Disponible a: <https://github.com/daniel-luberda/DLToolkit.Forms.Controls/blob/master/LICENSE>. [Consulta: 15-Mar-2019].

[64] “Top 10 Apache License Questions Answered.” [En línia]. Disponible a: <https://resources.whitesourcesoftware.com/blog-whitesource/top-10-apache-license-questions-answered>. [Consulta: 15-Mar-2019].

[65] J. Montemagno, *Pull To Refresh a ScrollView or ListView in Xamarin.Forms: jamesmontemagno/Xamarin.Forms-PullToRefreshLayout*. 2019. [En línia]. Disponible a: <https://github.com/jamesmontemagno/Xamarin.Forms-PullToRefreshLayout>. [Consulta: 15-Mar-2019].

[66] J. Montemagno, *Read and Write Settings Plugin for Xamarin and Windows: jamesmontemagno/SettingsPlugin*. 2019. [En línia]. Disponible a: <https://github.com/jamesmontemagno/SettingsPlugin/blob/master/LICENSE>. [Consulta: 15-Mar-2019].

[67] “NECADA,” *inLab FIB*, 17-Apr-2015. [En línia]. Disponible a: <https://inlab.fib.upc.edu/ca/necada>. [Consulta: 15-Mar-2019].

[68] “Fonts for software and hardware solutions.” [En línia]. Disponible a: <http://catalog.monotype.com/?keyword=gill%20sans%20MT>. [Consulta: 08-Apr-2019].

[69] “Gill Sans® Font Family,” *Fonts.com*. [En línia]. Disponible a: <https://www.fonts.com/font/monotype/gill-sans>. [Consulta: 08-Apr-2019].

[70] “Terms of Service | PlusPNG.” [En línia]. Disponible a: <http://pluspng.com/tos.php>. [Consulta: 08-Apr-2019].

[71] «Calculo GEI CeroCO2.» [En línia]. Disponible a: <https://www.ceroco2.org/calculadoras/home>. [Consulta: 09-Oct-2018]. (S’ha consultat altres calculadores d’aquest estil i els resultats eren similars)